Kernel Methods for Machine Learning with Life Science Applications

Abrahamsen, Trine Julie

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Kernel Methods for Machine Learning with Life Science Applications

Trine Julie Abrahamsen

Kongens Lyngby 2013
IMM-PhD-2013-299

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Building 303B, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253031, Fax +45 45882673
compute@compute.dtu.dk
www.compute.dtu.dk IMM-PhD-2013-299

Summary (English)

Kernel methods refer to a family of widely used nonlinear algorithms for machine learning tasks like classification, regression, and feature extraction. By exploiting the so-called kernel trick straightforward extensions of classical linear algorithms are enabled as long as the data only appear as innerproducts in the model formulation. This dissertation presents research on improving the performance of standard kernel methods like kernel Principal Component Analysis and the Support Vector Machine. Moreover, the goal of the thesis has been two-fold.

The first part focuses on the use of kernel Principal Component Analysis for nonlinear denoising. In this context stable solution of the inverse and inherently ill-posed pre-image problem constitutes the main challenge. It is proposed to stabilize the estimation by augmenting the cost function with either an ℓ_1- or ℓ_2-norm penalty, and solution schemes are derived for both approaches. The methods are experimentally validated on several biomedical data sets. Furthermore, frameworks for exploiting label information for improved denoising in the semisupervised case are proposed.

The second part of the thesis examines the effect of variance inflation in kernel methods. Variance inflation occurs in high-dimensional problems when the training data are insufficient to describe the entire signal manifold. Thereby leading to a potential mismatch between the subspaces spanned by the training and test data, respectively. It is shown how this effect extends from linear models to kernel learning, and means for restoring the generalizability in both kernel Principal Component Analysis and the Support Vector Machine are proposed. Viability is proved on a wide range of benchmark machine learning data sets.

Resumé (Danish)

Denne afhandling omhandler kernelmetoder til ikke-lineær dataanalyse. Kernelmetoder er en fællesbetegnelse for algoritmer, der benytter det såkaldte kerneltrick til at formulere ikke-lineære udvidelser af klassiske lineære algoritmer. Dette kan gøres så længe at data kun indgår som indreprodukter i den lineære model. Overordnet har denne afhandling to hovedmål. Den første del omhandler stabil støjreduktion ved kernel Principal Komponent Analyse hvorefter variansinflations problemet undersøges i relation til kernellæring.

Når kernel Principal Component Analysis anvendes til støjreduktion, er løsning af det inverse *ill-posed pre-image* problem essentielt. Stabil pre-image estimering udgør i denne forbindelse den største udfordring. Denne afhandling præsenterer nye pre-image algoritmer til forbedret støjreduktion ved at introducere henholdsvis ℓ_1- og ℓ_2-norm regularisering. Eksperimenter på håndskrevne tal samt biomedicinske datasæt illustrerer effekten af de nye estimatorer. Derudover introduceres metoder til at forbedre støjreduktionen, når klasse information er tilgængelig for en del af dataen.

Den anden del af denne afhandling omhandler varians-inflation i kernelmetoder. Varians-inflation kan forekomme i høj-dimensionale problemer, når mængden af træningsdata er utilstrækkelig til at repræsentere signalmanifolden. Dette medfører et muligt mismatch mellem underrummet udspændt af henholdsvis trænings- og testdata. I denne afhandling vises det, hvordan varians problemet forefindes i kernelalgoritmer, og metoder til at korrigere for det forøget variansestimat i både kernel Principal Komponent Analyse og Support Vektor Maskiner præsenteres. Standard *machine learning* datasæt anvendes til at illustrere, hvordan de foreslåede algoritmer gendanner generaliserbarheden.

Preface

This thesis was prepared at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark (DTU) in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering.

The thesis consists of a summary report and a collection of six published scientific papers and two papers currently under review elsewhere. The work was carried out between 2009 and 2013

Lyngby, 2-April-2013

Trine Julie Abrahamsen

List of Publications

Papers included in the thesis

[A] Trine Julie Abrahamsen and Lars Kai Hansen. Regularized Pre-image Estimation for Kernel PCA De-noising. *Journal of Signal Processing Systems*, 65:403–412, 2011. doi:10.1007/s11265-010-0515-4.

[B] Trine Julie Abrahamsen and Lars Kai Hansen. Sparse Non-linear Denoising: Generalization Performance and Pattern Reproducibility in Functional MRI. *Pattern Recognition Letters*, 32(15):2080 – 2085, 2011. doi:10.1016/j.patrec.2011.08.006.

[C] Peter Mondrup Rasmussen, Trine Julie Abrahamsen, Kristoffer Hougaard Madsen, and Lars Kai Hansen. Nonlinear Denoising and Analysis of Neuroimages with Kernel Principal Component Analysis and Pre-image Estimation. *NeuroImage*, 60(3):1807 – 1818, 2012. doi:10.1016/j.neuroimage.2012.01.096.

[D] Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. Information-based Kernel PCA Denoising by Semi-supervised Manifold Learning. *Pattern Recognition Letters*, Under review, 2013.

[E] Trine Julie Abrahamsen and Lars Kai Hansen. A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis. *Journal of Machine Learning Research*, 12:2027–2044, 2011. Available from: `http://jmlr.csail.mit.edu/papers/v12/abrahamsen11a.html`.

[F] Trine Julie Abrahamsen and Lars Kai Hansen. Restoring the Generalizability of SVM Based Decoding in High Dimensional Neuroimage Data.

In Georg Langs, Irina Rish, Moritz Grosse-Wentrup, and Brian Murphy, editors, *Machine Learning and Interpretation in Neuroimaging*, Lecture Notes in Computer Science, pages 256–263. Springer Berlin Heidelberg, 2012.
doi:10.1007/978-3-642-34713-9_32.

[G] Trine Julie Abrahamsen and Lars Kai Hansen. Variance Inflation in High Dimensional Support Vector Machines. *Pattern Recognition Letters*, Accepted, 2013.

[H] Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. A Randomized Heuristic for Kernel Parameter Selection with Large-scale Multi-class Data. In *Proceedings of 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2011.
doi:10.1109/MLSP.2011.6064582.

Additional papers not included in the thesis

[I] Trine Julie Abrahamsen, and Lars Kai Hansen. Restoring the Generalizability of SVM based Decoding in High Dimensional Neuroimage Data. *Neural Information Processing Systems (NIPS) Workshop on Machine Learning and Interpretation in Neuroimaging*, 2011.

[J] Trine Julie Abrahamsen, and Lars Kai Hansen. Sparse non-linear denoising of fMRI: Performance and pattern reproducibility. *Neural Information Processing Systems (NIPS) Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions*, 2010.

Acknowledgements

The work contained in this thesis would not have been possible without the help and support of many people. First of all, I would like to express my gratitude to Professor Lars Kai Hansen for his knowledge, advice, and encouragement. His support over the past years has been invaluable. Along the same lines I am thankful for the support by Associate Professor Ole Winther. I would like to thank all past and present members of the Cognitive Systems group for creating an inspiring and fun environment. A special thanks goes to Jacob S. Vestergaard, Morten Arngren, Toke J. Hansen, Bjarne Ø. Wahlgreen, and Tue Herlau. Furthermore, I am thankful for the collaboration and fruitful discussions with my co-authers, Toke J. Hansen, Peter M. Rasmussen, Kristoffer H. Madsen, and Lars K. Hansen.

During my studies I had the privilege to visit Professor Bin Yu's group at the Department of Statistics at University of California, Berkeley. I wish to thank Professor Bin Yu for her hospitality and advise, and every member of the Yu-group for making my stay unforgettable.

I am grateful for the financial support from Otto Mønsteds Fond, Zonta Danmark, and the Danish Ministry of Science and Education without whom my external stay would not have been possible. Finally, the work in this thesis has been supported in part by the Danish Lundbeckfonden through the Center for Integrated Molecular Brain Imaging, while the Danish Research Centre for Magnetic Resonance, Hvidovre Hospital, Denmark kindly provided fMRI data.

Finally, but not least I am eternally grateful for the endless support i have received from family and friends over the past years.

Nomenclature

Acronyms and Abbreviations

ALOI	Amsterdam Library of Object Images 27
CV	Cross-Validation 20
fMRI	functional Magnetic Resonance Imaging 38
GP	Gaussian Process 8
GPS	Generalized Path Seeking 26
i.i.d.	independent and identically distributed 15
KKT	Karush-Kuhn-Tucker 18
KR	Kent Ridge biomedical data set repository 34
LASSO	Least Absolute Shrinkage and Selection Operator 30
LEM	Laplacian Eigenmaps 13
LLE	Locally Linear Embedding 24
LOO	Leave-One-Out 39
MAP	Maximum A Posteriori 26
MDS	Multidimensional Scaling 24
MEB	Minimum Enclosing Ball 20
ML	Maximum Likelihood 30
MSE	Mean Squared Error 38
NPAIRS	Nonparametric, Prediction, Activation, Influence, Reproducibility, re-Sampling 38

Mathematical Notation and Fixed Symbols

x	Scalar
$\boldsymbol{x}$	Coloumn vector
$\boldsymbol{X}$	Matrix
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	Innerproduct of $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{x}^\top$ or $\boldsymbol{X}^\top$	Vector or matrix transpose
$\bar{\boldsymbol{x}}$	The mean of $\boldsymbol{x}$
$\widetilde{\boldsymbol{x}}$	$\boldsymbol{x}$ after centering
$\mathcal{X}$	Input space
$\mathcal{H}$	Reproducing Kernel Hilbert Space
$\varphi(\cdot)$	Map from input space to Reproducing Kernel Hilbert Space
$k(\cdot, \cdot)$	Kernel function, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle$
$\boldsymbol{K}$	Kernel (or Gram) matrix, $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
γ	Scale in the Gaussian kernel, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\right)$
C	Slack parameter in the Support Vector Machine
q	Dimensionality of principal subspace in (kernel) PCA
$P_q \varphi(\boldsymbol{x})$	Projection of $\varphi(\boldsymbol{x})$ onto the subspace spanned by the first q eigenvectors
N	The number of observations in a data set
D	Dimensionality of a vector space
$\mathbb{R}/\mathbb{R}_+$	The set of real numbers/postive real numbers
$\mathbb{N}_+$	The set of positive natural numbers $\{1, 2, \ldots\}$

Contents

Chapter 1

Introduction

This opening chapter serves as a general introduction to the work contained in the thesis.

Section 1.1 comprises a motivation for the use of kernel methods and presents the challenges addressed in remainder of the thesis, while Section 1.2 gives a chapter-by-chapter overview, including a brief summary of the scientific papers.

1.1 Motivation

Kernel based learning algorithms can be seen as nonlinear extensions of classical linear algorithms. The common feature of these methods is the way the nonlinearity is introduced. In general, kernel methods are comprised of two steps. Initially, the data are mapped by a potentially nonlinear function into the kernel embedding or feature space, and then secondly a standard linear method is implemented in this new space. The nonlinearity is controlled by the mapping function, which can be implicitly defined by the choice of kernel function. The main cause to kernel methods widespread use stems from the fact that as long as the data only appear as innerproducts, all calculations can be done without explicitly representing the data in the high-dimensional feature space. This is due to the kernel trick, stating that innerproducts in feature space can be evaluated in terms of kernel evaluations in input space.

The list of linear methods for which a nonlinear kernelized version has been developed is long and range from unsupervised pattern recognition methods like kernel Principal Component Analysis (PCA) through semisupervised methods to supervised classification and regression algorithms like the Support Vector Machine (SVM). In recent years advances in data collection, storage and computing resources have facilitated an increasing popularity of kernel methods due to the computational simplicity of these methods, and so applications among others include object recognition, text categorization, time series prediction, gene expression, and DNA analysis.

The aim of the work contained in this thesis has been two-fold. While the first part focuses on improved nonlinear denoising by kernel PCA, the second part of the thesis relates to the loss of generalizability in kernel learning due to variance inflation caused by incomplete learning from small samples in high-dimensional spaces. The proposed methods are experimentally validated on handwritten digits from the USPS data base [Hul94], neuroimaging data in the form of functional Magnetic Resonance Imaging (fMRI), or benchmark machine learning data sets from the University of California, Irvine machine learning repository (UCI) [FA10] and the Kent Ridge biomedical data set repository (KR)[1].

Denoising by kernel PCA can be thought of in three steps. First, the data are mapped to the high-dimensional feature space, where linear PCA is then implemented implicitly. The final step consists of mapping the hopefully less noisy projection back to the original input space. This last step is known as the preimage problem. For many choices of kernel embedding the inverse problem is in-

[1] Available at `http://datam.i2r.a-star.edu.sg/datasets/krbd/`

herently ill-posed and suffers from lack of stability in the nonlinear regime. This thesis proposes means for improved denoising by kernel PCA by introducing stable estimation schemes for solving the pre-image problem. ℓ_2-norm regularization is introduced to stabilize the estimation while ℓ_1-norm regularization is used for improved visual quality when the reconstruction is expected to be sparse. Finally, a semi-supervised scheme is introduced for problems where label information is available on a subset of the observations. The work related to denoising by kernel PCA is contained in the papers [AH11b, AH11c, RAMH12, HAH13].

The problem of variance inflation in high-dimensional problems relative to the sample size is well known in linear methods. When the training data are insufficient to describe the signal manifold, a mismatch between the subspaces spanned by the training and test data respectively can occur. This will lead to an overestimate of the variance on the training data, or more generally that the training and test sets will follow different probability laws. The consequence is poor generalization to unseen test data in these problems. Inspired by the work on linear PCA in [KHS01a], this thesis investigates the extent of variance inflation in kernel methods and suggests approaches to restore the generalizability of kernel methods like kernel PCA and SVMs. The work regarding the variance inflation problem in kernel learning is presented in [AH11a, AH12, AH13].

1.2 Thesis Outline and contributions

In addition to the current chapter the thesis consists of four introductory chapters, six published papers, and two papers currently under review elsewhere. The papers are found in the appendices and constitute the main contribution of the thesis. The introductory chapters aim at providing an overview of kernel methods and introduce kernel PCA denoising, the SVM, and the variance inflation problem in greater detail. In summary, the remainder of the thesis is structured as follows:

Chapter 2, Kernel Methods, provides a foundation for the rest of the thesis. The general concept of kernels is introduced followed by an elaboration on kernel PCA and the SVM.

Chapter 3, Denoising by Kernel PCA, presents the challenges of denoising by kernel PCA and reviews the current estimation schemes for solving the pre-image problem. Furthermore, the contributions of this thesis are summarized.

Chapter 4, Variance Inflation, focuses on the cause of variance inflation in small sample high-dimensional learning. The concepts are presented in

terms of standard linear models prior to an exposition of the contributions of this thesis regarding variance inflation in kernelized methods.

Chapter 5, Conclusions, recapitulates the main findings presented in the thesis.

Paper A, Regularized Pre-image Estimation for Kernel PCA De-noising, [AH11b], addresses the instability issues in current estimators for the pre-image problem in denoising by kernel PCA. Novel estimators are proposed by augmenting the pre-image reconstruction cost function with either an ℓ_2- or ℓ_1-norm penalty term in order to improve the stability or sparseness of the reconstruction, respectively. Experimental results on handwritten digits illustrate how ℓ_2-norm regularization stabilizes the pre-image estimate with very little sacrifice in terms of denoising efficiency, while ℓ_1-norm regularization leads to improved visual quality of the reconstruction.

Paper B, Sparse Non-linear Denoising: Generalization Performance and Pattern Reproducibility in Functional MRI, [AH11c], elaborates on the sparse pre-image reconstuction originally presented in [AH11b]. fMRI data are used to illustrate how sparse estimation leads to both improved brain state decoding accuracy and higher reproducibility of the reconstruction.

Paper C, Nonlinear Denoising and Analysis of Neuroimages with Kernel Principal Component Analysis and Pre-image Estimation, [RAMH12], focuses on the use of kernel PCA denoising in neuroimaging. Extensive experiments are used to show that pre-image estimation significantly improve the reproducibility of the brain maps while maintaining the predictive performance. Additionally, it is illustrated how pre-image estimation allows for exploration of the nonlinear signal manifold between experimentally defined brain states.

Paper D, Information-based Kernel PCA Denoising by Semi-supervised Manifold Learning, [HAH13], constitutes the last paper of the thesis related to kernel PCA denoising. Two approaches for exploiting label information in order to improve the denoising in semisupervised problems are proposed. First, the original kernel PCA formulation is augmented by a loss term for the labeled data, leading to an iterative algorithm for finding orthonormal components biased by the class labels. Secondly, a fixed-point iteration scheme for solving the pre-image problem for a manifold warped Reproducing Kernel Hilbert Space (RKHS) is derived. The effect of the proposed methods is illustrated on an image classification problem, where it is shown that incorporating label information decreases the sensitivity to the choice of kernel hyperparameter and improves the denoising as measured by the Mean Squared Error (MSE),

thereby indicating that a more descriptive manifold representation has been achieved.

Paper E, A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis, [AH11a], has two main contributions. First, a computational efficient approximate Leave-One-Out (LOO) scheme is proposed for restoring the generalizability in small sample high-dimensional linear PCA. Secondly, it is shown how the variance inflation problem extends from linear PCA to kernel PCA and a nonparametric scheme for renormalization of the kernel embedded projections are suggested. Viability of the proposed methods are demonstrated on handwritten digit data and on fMRI brain state decoding.

Paper F, Restoring the Generalizability of SVM Based Decoding in High Dimensional Neuroimage Data, [AH12], investigates the effect of variance inflation in the SVM. The lack of generalizability is found to be pronounced in ill-posed imbalanced problems. Inspired by [AH11a], a nonparametric renormalization procedure for restoring the variation is proposed and experimentally validated on handwritten digits data and a fMRI brain state decoding problem. The experiments suggest that applying the proposed renormalization scheme not only improves performance but also leads to a more nonlinear optimal kernel embedding in the neuroimaging problem.

Paper G, Variance Inflation in High Dimensional Support Vector Machines, [AH13], elaborates on the variance inflation problem in supervised kernel learning initially noted in [AH12], and introduces a LOO based framework for problems where the training and test data have different prior class distributions. The renormalization schemes are applied to a wide range of benchmark data sets with varying N/D ratio, illustrating that it is indeed possible to restore the performance in ill-posed problems.

Paper H, A Randomized Heuristic for Kernel Parameter Selection with Large-Scale Multi-Class Data, [HAH11], deals with the problem of hyperparameter selection in kernel algorithms where Cross-Validation (CV) is infeasible due to the problem size. A novel heuristic for finding the optimal hyperparameter is suggested based on fitting a Minimum Enclosing Ball (MEB) to the class means in the RKHS. Experiments on an image data set containing multiple classes show that the proposed randomized approach leads to both improved performance and very competitive time complexities compared to other distance metrics in the RKHS.

Chapter 2

Kernel Methods

The basic concept in kernel learning is to exploit the so-called kernel trick to formulate nonlinear extensions of classical linear models by implicitly mapping the data to a high-dimensional kernel induced feature space or Reproducing Kernel Hilbert Space. Thereby forming a framework where nonlinearities are easily introduced as long as the data only appear as innerproducts in the model formulation.

This chapter consists of a concise review of kernel methods. In this context Section 2.1 serves as a general introduction to the properties of kernels, while Section 2.2 reviews kernel Principal Component Analysis. Finally, Section 2.3 provides an exposition of the Support Vector Machine. The novel contributions of this thesis are highlighted at the end of each section.

2.1 Introduction to Kernels

Linear methods constitute a trusted workhorse in todays data analysis and their theory and applications have been widely studied. However, many real world problems can not be described solely in terms of linear relations, and hence the need for nonlinear analysis methods arise.

Recent years have seen an increase in the popularity of estimation and learning methods within machine learning which are based on positive semidefinite (psd) kernels. These methods form a large group of nonlinear extensions to classical linear algorithms denoted by the collective term kernel methods. The kernel trick, and hence kernel based learning was initially introduced in [ABR64]. In [Vap79, Vap95, Vap98] V. Vapnik derived the Support Vector Machine (SVM) for binary classification and kernel based versions of other supervised as well as unsupervised algorithms like kernel Principal Component Analysis (PCA) [SSM98] and kernel Fisher Discriminant [MRW$^+$99] quickly followed. Other branches of kernel methods include Gaussian Processes (GPs) which, unlike the SVM that is rooted in statistical learning theory, was developed from the theory of stochastic processes. Since GPs are beyond the scope of this thesis the reader is referred to, e.g., [WR96, Mac98, Wil98] for reviews.

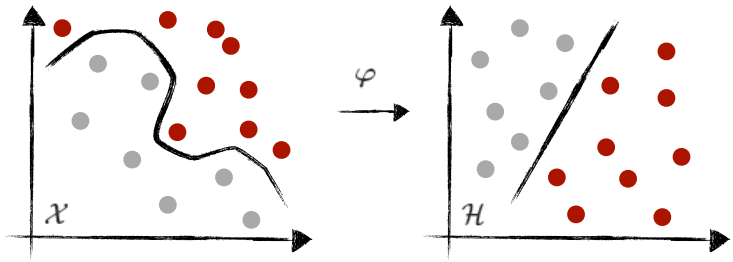


Figure 2.1: Illustration of the basic idea in kernel methods. By choosing a suitable feature space map, the data are embedded in a RKHS, $\mathcal{H}$, where relations are linear. The mapping function, φ, can be defined implicitly by the choice of kernel.

Kernel methods essentially consist of two steps. Initially, the data are mapped from the original input space, $\mathcal{X}$, to a RKHS, $\mathcal{H}$. In the following, RKHS and feature space will be used interchangeably to describe the kernel induced embedding. Secondly, a standard linear method is applied in this new space. The hope of kernel methods is that there exists an embedding in which the data consist of linear patterns (see Figure 2.1 for an illustration). The mapping function, and hence the representation, can be chosen indirectly by the choice

of kernel. As discussed in the following paragraphs, the so-called kernel trick implies that the complexity of the transformed problem depends solely on the size of the training set and not on the dimensionality of new space, which leads kernel methods to form an efficient framework for working in arbitrarily high- (or even infinite-) dimensional feature spaces. [Bur98, Her01, SS01, STC04, HSS08]

In the following some fundamental definitions will be outlined briefly as a formal introduction of kernel learning. A kernel function, $k(\cdot,\cdot)$, is a function defined on $\mathcal{X} \times \mathcal{X}$ that for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ satisfy [STC04]

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle \,, \tag{2.1}$$

where $\varphi : \mathcal{X} \mapsto \mathcal{H}$ is a possibly nonlinear map from the $D_{\mathcal{X}}$-dimensional input space, $\mathcal{X}$, to the $D_{\mathcal{H}}$-dimensional RKHS associated with the kernel. So the kernel function computes the innerproduct of the images of the data under the embedding, φ.

The definition given in Equation (2.1) is also known as the kernel trick which states that innerproducts in $\mathcal{H}$ can be computed in terms of kernel evaluations in $\mathcal{X}$. The kernel trick is the crucial part in making kernel methods feasible, since it implies that the data never have to be represented explicitly in the RKHS as long as the data only appear as innerproducts in the model formulation. It turns out that many linear methods can indeed be reformulated using only innerproducts, and so kernel methods form a straightforward framework for modeling nonlinear signal manifolds in a computationally efficient manner. [SMB+99, HTF01, Bis06]

Definition 2.1 (Kernel Matrix) Given a kernel, k, and observations, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \mathcal{X}$, then the $N \times N$ matrix

$$\boldsymbol{K} = \big(k(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)_{ij} = \big(\, \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle \,\big)_{ij},$$

is the kernel (or Gram) matrix of k.

Definition 2.2 (Postive semidefinite matrix) A symmetric matrix, $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, satisfying

$$\sum_{i,j=1}^{N} c_i c_j \boldsymbol{K}_{ij} \geq 0 \quad \forall c_i, c_j \in \mathbb{R},$$

is called positive semidefinite (psd).

Since $\sum_{ij} c_i c_j \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle = \left\langle \sum_i c_i \varphi(\boldsymbol{x}_i), \sum_j c_j \varphi(\boldsymbol{x}_j) \right\rangle \geq 0$, all kernels of the form in Equation (2.1) are by definition psd for any choice of φ. [HSS08]

In early years, kernels were required to fulfill Mercer's Theorem [Mer09] in order to be valid

Theorem 2.3 (Mercer's Theorem) *A symmetric function $k(\cdot,\cdot)$ can be expressed as an innerproduct*

$$k(\boldsymbol{x},\boldsymbol{x}') = \langle\varphi(\boldsymbol{x}),\varphi(\boldsymbol{x}')\rangle$$

for some φ if and only if $k(\boldsymbol{x},\boldsymbol{x}')$ is positive semidefinite, i.e.:

$$k(\boldsymbol{x},\boldsymbol{x}') = \int\int k(\boldsymbol{x},\boldsymbol{x}')g(\boldsymbol{x})g(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{x}' \geq 0 \quad \forall g \in L_2(\mathcal{X})$$

or, equivalent

$$\boldsymbol{K} \text{ is positive semidefinite for all sets } \{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\}.$$

However, while all Mercer kernels indeed satisfy Equation (2.1) the opposite does not always hold [HSS08]. Hence, the requirements can be relaxed so that a kernel is considered valid if and only if it is symmetric and psd. When this is fulfilled, it is guaranteed that there exsists a map $\varphi : \mathcal{X} \mapsto \mathcal{H}$ such that Equation (2.1) holds and the feature space, $\mathcal{H}$, will have the structure of a RKHS. This is basically equivalent to $\boldsymbol{K}$ being an innerproduct matrix in some space. From the Moore-Aronszajn Theorem [Aro50] it follows that every psd kernel on $\mathcal{X}\times\mathcal{X}$ is associated with a unique RKHS and vice versa.

Table 2.1: Common choices of kernel functions, where $c,\gamma \in \mathbb{R}_+$ and $p \in \mathbb{N}_+$

Projective kernels	
Polynomial	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = c + \langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle^p$
Exponential	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp(\gamma\,\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle)$
Sigmoid (perceptron)	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \tanh(c+\gamma\,\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle)$
Radial Basis Function (RBF) kernels	
Gaussian	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\right)$
Laplacian	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\right)$
Multiquadratic	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sqrt{c+\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}$
Inverse multiquadratic	$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = 1/\sqrt{c+\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}$

The majority of the work contained in this thesis uses the Gaussian kernel given by $k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\right)$, where γ is the scale parameter controlling the nonlinearity of the map. A small value of γ relative to the square of the pairwise distances between the observations in $\mathcal{X}$ will lead to a very linear kernel function, and in the limiting case any kernelized method will approach its

linear equivalent since high order terms in the Taylor expansion of the Gaussian kernel become insignificant. On the contrary, a larger γ results in a more nonlinear kernel embedding, where $\boldsymbol{K} = \boldsymbol{I}$ in the limiting case, i.e., all observation become dissimilar. Thus, choosing the optimal kernel hyperparameter is not straightforward. The Gaussian kernel belongs to the family of RBF kernels, which are isotropic and have infinite support, making the dimensionality of the associated feature space infinite. Other commonly used kernel functions are listed in Table 2.1. Furthermore, there exists kernels for embedding general structures like sequences, trees, graphs, semantics of text, and for probabilistic graphical models. [HR11]

The kernel embedding can either be chosen explicitly by the map, φ, followed by construction of all innerproducts, or by choosing a valid kernel function, k, and thereby define the embedding implicitly. In the latter case, the kernel embedding, φ, can be viewed as representing each data point by a kernel shaped function centered on the observation itself. This is illustrated in Figure 2.2 for a Gaussian kernel function, and is essentially equivalent to representing all observations by their similarity in $\mathcal{X}$ with respect to the similarity measure defined by the kernel function.

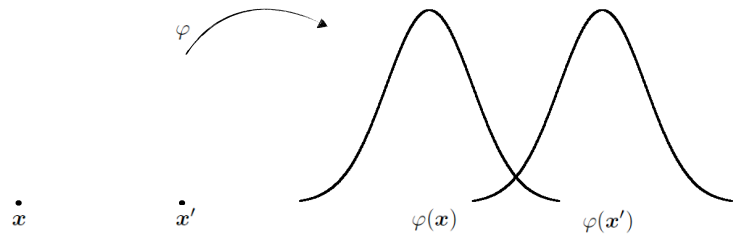


Figure 2.2: Illustration of the feature space mapping, φ, using a Gaussian kernel. Modified from [SS01].

The choice of kernel function and its hyperparameters is critical to the success of all kernel algorithms. Since any symmetric and psd function constitutes a valid kernel and most functions have one or more hyperparameters, the user is left with an immense amount of possible choices. This have led to the development of multiple kernel learning [WMC+01, CVBM02], where the idea is to only specify a family of kernels and then learn the kernel from the data. Thereby the risk of error is minimized in cases where the user does not have sufficient knowledge to select an appropriate kernel a priori. Multiple kernel learning is byond the scope of this thesis, but the reader is referred to, e.g., [GA11] for a review.

Many kernel methods eventually leads to an optimization problem in $\mathcal{H}$. From the representer theorem [KW71, SS01] it follows that allthough the problem is defined in the high- or infinite-dimensional RKHS, the solution always lies in the N-dimensional subspace defined by the span of the kernels centered on the N training observations.

Theorem 2.4 (Representer Theorem) *Given training data* $(\boldsymbol{x}_1, y_1)$, $\ldots, (\boldsymbol{x}_N, y_N) \in \mathcal{X} \times \mathbb{R}$, *let* $T : [0, \infty[\mapsto \mathbb{R}$ *be a strictly monotonic increasing function and* $R : (\mathcal{X} \times \mathbb{R}^2)^N \mapsto \mathbb{R} \cup \{\infty\}$ *an arbitrary loss function. Then each minimizer* $f \in \mathcal{H}$ *of the regularized risk:*

$$R\left((\boldsymbol{x}_1, y_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_N, y_N, f(\boldsymbol{x}_N)),\right) + T(||f||^2),$$

can be written on the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}),$$

where $\alpha_i \in \mathbb{R},\ \forall 1 \leq i \leq N$.

The representer theorem effectively reduces the original minimization problem to that of finding the optimal coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$, whereby the complexity is reduced from possibly infinite to N-dimensional. Furthermore, many of the α_i's will often be zero for suitable choices of loss functions. [HSS08]

As a final remark, most algorithms assume that the data are centered in $\mathcal{H}$, i.e., $\tilde{\varphi}(\boldsymbol{x}) = \varphi(\boldsymbol{x}) - \bar{\varphi}$, where $\bar{\varphi}$ is the mean of the data in $\mathcal{H}$. However, due to the possible infinite dimensionality of $\mathcal{H}$, the centering operation cannot always be performed explicitly. Instead a centered version of the kernel matrix can be derived: (cf. [SSM98])

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \frac{1}{N}\mathbf{1}_{NN}\boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\mathbf{1}_{NN} + \frac{1}{N^2}\mathbf{1}_{NN}\boldsymbol{K}\mathbf{1}_{NN},$$

where $\mathbf{1}_{NN}$ is a $N \times N$ matrix of ones. Suppose M test observations are given, then the centered kernel matrix $\widetilde{\boldsymbol{K}}_{test} \in \mathbb{R}^{M \times N}$ is found as

$$\widetilde{\boldsymbol{K}}_{test} = \boldsymbol{K}_{test} - \frac{1}{N}\mathbf{1}_{NM}^T\boldsymbol{K} - \frac{1}{N}\boldsymbol{K}_{test}\mathbf{1}_{NM} + \frac{1}{N^2}\mathbf{1}_{NM}^T\boldsymbol{K}\mathbf{1}_{NN}.$$

For a more thorough review of kernel methods please refer to, e.g., [SS01, STC04, HSS08].

2.2 Kernel Principal Component Analysis

Linear PCA [Hot33] is a well-trusted unsupervised method for extracting the main modes of variation and is often used for dimensionality reduction, data compression, feature extraction, denoising, or visualization. In order to achieve analogue functionalities in nonlinear manifolds, several techniques have been developed including kernel PCA [SSM98, SS01], Locally Linear Embedding (LLE) [RS00, SR03], Laplacian Eigenmaps (LEM) [BN03], Isomap [T$^+$98, TDSL00], and Semidefinite Embedding (SDE) [WSS04, WS04].

Kernel PCA was originally introduced in [SSM98] as a generalization of linear PCA to nonlinear manifolds, and it has proven a powerful tool for nonlinear dimensionality reduction, feature extraction, or denoising. Common for all applications is that the goal is to only retain a subset of the principal components. As opposed to other nonlinear feature extraction methods, kernel PCA does not require nonlinear optimization but instead the solution of an eigenvalue problem. [BBM08]

Analogue to linear PCA, the aim of kernel PCA is to project the data onto an orthonormal basis that maximizes the explained variance. However, this should hold in $\mathcal{H}$ not $\mathcal{X}$. The basis is determined by the leading eigenvectors of the covariance matrix in the RKHS, given by $\boldsymbol{C} = \frac{1}{N}\sum_{n=1}^{N}\tilde{\varphi}(\boldsymbol{x}_n)\tilde{\varphi}(\boldsymbol{x}_n)^\top$. The magnitude of the i'th eigenvalue, λ_i, measures the amount of variation in the direction of the corresponding eigenvector, $\boldsymbol{v}_i$, which is also known as the i'th Principal Component (PC). Analogue to linear PCA the i'th PC can be found as the normal direction that maximizes the variance of the projection while being orthogonal to all previous PCs. This can be formulated as a quadratic optimization problem:

$$\begin{aligned} \underset{\boldsymbol{v}_i \in \mathcal{H}}{\text{maximize}} \quad & \boldsymbol{v}_i^\top \boldsymbol{C} \boldsymbol{v}_i \\ \text{s.t} \quad & \|\boldsymbol{v}_i\|^2 = 1 \\ & \sum_{n=1}^{i-1} \langle \boldsymbol{v}_n, \boldsymbol{v}_i \rangle^2 = 0. \end{aligned} \tag{2.2}$$

Since $\boldsymbol{C}$ is a covariance matrix, and hence psd, the problem is known to be convex, and similar to linear PCA it can be expressed as an eigenvalue equation

$$\lambda_i \boldsymbol{v}_i = \boldsymbol{C}\boldsymbol{v}_i, \tag{2.3}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ are the ordered eigenvalues and $\boldsymbol{v}_i \in \mathcal{H}\backslash\{\boldsymbol{0}\}$ are the corresponding eigenvectors of $\boldsymbol{C}$. Since $\boldsymbol{C}\boldsymbol{v}_i = \frac{1}{N}\sum_{n=1}^{N}\tilde{\varphi}(\boldsymbol{x}_n)\langle\tilde{\varphi}(\boldsymbol{x}_n), \boldsymbol{v}_i\rangle$, it follows that all solutions $\boldsymbol{v}_i$ with $\lambda_i > 0$, must lie in the span of the centered training images, i.e., $\boldsymbol{v}_i = \sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\boldsymbol{x}_n)$. By substitution and simple

manipulations it follows that the eigenvalue problem in Equation (2.3) can be reformulated as

$$N\lambda_i \boldsymbol{\alpha}_i = \widetilde{\boldsymbol{K}}\boldsymbol{\alpha}_i.$$

The $\boldsymbol{\alpha}$'s are scaled by requiring that the corresponding $\boldsymbol{v}$-vectors are normalized in $\mathcal{H}$, from which it follows that $\boldsymbol{\alpha}_i \leftarrow \frac{\boldsymbol{\alpha}_i}{N\lambda_i}$. For a full derivation of kernel PCA the reader is referred to, e.g., [SSM98, STC04].

The projection of $\tilde{\varphi}(\boldsymbol{x})$ onto the principal direction, $\boldsymbol{v}_i$, can now be computed as

$$\beta_i = \tilde{\varphi}(\boldsymbol{x})^\top \boldsymbol{v}_i = \sum_{n=1}^{N} \alpha_{in} \langle \tilde{\varphi}(\boldsymbol{x}), \tilde{\varphi}(\boldsymbol{x}_n) \rangle = \sum_{n=1}^{N} \alpha_{in} \tilde{k}(\boldsymbol{x}, \boldsymbol{x}_n), \tag{2.4}$$

and the projection onto the subspace spanned by the first q eigenvectors is similarly given by

$$P_q \varphi(\boldsymbol{x}) = \sum_{i=1}^{q} \beta_i \boldsymbol{v}_i + \bar{\varphi} = \sum_{i=1}^{q} \beta_i \sum_{n=1}^{N} \alpha_{in} \tilde{\varphi}(\boldsymbol{x}_n) + \bar{\varphi} = \sum_{n=1}^{N} \tilde{\xi}_n \tilde{\varphi}(\boldsymbol{x}_n) + \bar{\varphi}, \tag{2.5}$$

where $\tilde{\xi}_n = \sum_{i=1}^{q} \beta_i \alpha_{in}$.

In the context of this thesis, kernel PCA is used for denoising when the signal manifold is expected to be nonlinear. For denoising purposes, it is necessary to estimate the inverse mapping, φ^{-1}, in order to reconstruct the hopefully less noise observation in the original input space after kernel PCA has been performed. This problem is known as the pre-image problem, and will be introduced in Chapter 3. Furthermore, kernel PCA is used to investigate the variance inflation problem in the kernel induced feature space as described in Chapter 4.

Contributions of this thesis

In [HAH13] (see Appendix D) two semisupervised methods for incorporating label information in kernel PCA denoising is proposed. The aim of the first method is to exploit the available class labels to devise a more descriptive manifold representation and thereby improve performance. The approach elaborates on the work of [WHMH10], who originally presented a semisupervised solution for the leading eigenvector. In [HAH13] the kernel PCA formulation in Equation (2.2) is augmented with a loss term for the labeled data. Since the extended objective can no longer be formulated as an eigenvalue problem, an iterative scheme for finding several PCs biased towards the labeled data are developed by exploiting ideas originally presented in [GGvM89].

Experiments on synthetic data as well as images from the Amsterdam Library of Object Images (ALOI) data base [GBS05] indicate that the label informed kernel PCA scheme improves performance in terms of a lower Mean Squared Error (MSE) compared to using the standard unsupervised kernel PCA formulation. By varying the ratio of labeled versus unlabeled observations it is found that for a fixed sample size, a higher fraction of labeled data improves performance. At the same time the experiments suggest that for a fixed number of labeled samples adding additional unlabeled data also significantly lowers the MSE. Finally, the semisupervised approach is found to be more robust to the choice of kernel hyperparameter. The second framework proposed in [HAH13] relates to the pre-image problem and will be described in Chapter 3.

2.3 Support Vector Machines

Although the idea of the SVM was to some extend introduced by [Vap79] it took more than 15 years for the subject to receive proper attention. Following the publication of [Vap95, Vap98], SVMs became an increasing area of research, and even though the SVM was originally proposed as a computationally powerful supervised learning algorithm for binary classification problems, it was quickly extended to multi-class problems and regression [BGV92, Bur98, SSWB00, SS04].

The SVM is rooted in statistical learning theory as developed in [Vap95], exploiting Vapnik-Chervonenkis (VC) theory [VC71] and the VC dimension in particular as a measure of a algorithms capacity to learn from data. Even though capacity control is used for model selection, the theory is also closely related to regularization. In general, an algorithms capacity can be described as its ability to balance the accuracy achieved on a finite amount of training data and the ability to generalize to any unseen data without error. For more details on the SVMs relation to risk minimization and VC dimension, please refer to, e.g., [Bur98].

Some of the favorable properties of the SVM include that the optimization problem turns out to be convex, that the decision function is fully specified in terms of innerproducts, making the kernel trick applicable, and that it is robust with respect to the choice of parameters. This has led SVMs to be empirically superior to other algorithms in a broad variety of fields. [SS01, CST00, STC04, BEWB05]

Now, let $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, $\boldsymbol{x} \in \mathbb{R}^D$, $y \in \{+1, -1\}$ be a set of independent and identically distributed (i.i.d.) training data generated from an unknown

probability distribution $P(\boldsymbol{x}, y)$. The aim of any binary classifier is to learn a decision function, $f : \mathbb{R}^D \mapsto \{+1, -1\}$, from the data, $\mathcal{D}$, such that f correctly classify new unseen test data drawn from $P(\boldsymbol{x}, y)$. Initially, it assumed that the two classes are separable in some $\mathcal{H}$.

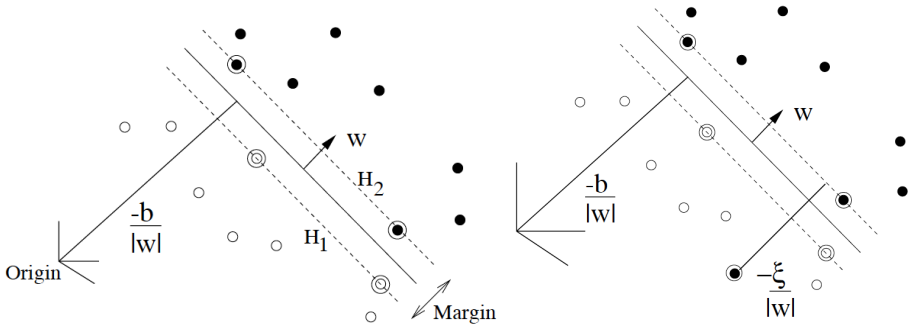


Figure 2.3: Left: Illustration of the optimal separating hyperplane for binary classification. The two hyperplanes defining the margin are denoted H_1 and H_2, while $\boldsymbol{w}$ and b are given in Equation (2.6). Right: In the SVM formulation slack parameters, ξ_i, are introduced, allowing for the hard margin constraints to be violated. The support vectors are highlighted in both panels. From [Bur98].

The SVM defines the optimal decision function as the maximum margin hyperplane that separates the two classes. Any hyperplane in some kernel induced feature space, $\mathcal{H}$, can be formulated as

$$\{\varphi(\boldsymbol{x}) \in \mathcal{H} \ : \ \langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + b = 0\} \quad \boldsymbol{w} \in \mathcal{H}, \boldsymbol{b} \in \mathbb{R}, \tag{2.6}$$

where $\boldsymbol{w}$ is the weight vector and b is the bias term. As indicated in the left panel of Figure 2.3, $\boldsymbol{w}$ is normal to the hyperplane, and $|b|/||\boldsymbol{w}||$ is the orthogonal distance from the hyperplane to the origin. $\langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + b$ is the signed distance from $\boldsymbol{x}$ to the hyperplane and will be referred to as the decision value. Hence, the classifiers decision function can be expressed as

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \operatorname{sgn}\Big(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + b \Big). \tag{2.7}$$

In most cases, several hyperplanes can be used to separate the two classes. The optimal maximum margin separating hyperplane, is defined by the two hyperplanes with maximum distance between them that separates the classes and have no points between. The region bounded by the two hyperplanes is called the margin. This can also be seen in the left panel of Figure 2.3, where the two hyperplanes defining the margin are denoted H_1 and H_2, respectively. By exploiting that the classification function is invariant to rescaling, these

hyperplanes can be expressed as

$$\langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + b = 1 \text{ and } \langle \boldsymbol{w}, \varphi(\boldsymbol{x}) \rangle + b = -1,$$

respectively. When considering the hard margin case, there can be no data between the two hyperplanes. Hence, it further follows that

$$y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i) \rangle + b) \geq 1, \ \forall i, \tag{2.8}$$

which is known as the canonical representation of the decision hyperplane [STC04, Bis06]. From geometry it follows that the width of the margin is given by $2/||\boldsymbol{w}||$. So the maximum margin separating hyperplane can be found by minimizing $||\boldsymbol{w}||$ subject to that there are no observations within the margin, which can be cast into a quadratic optimization problem

$$\begin{aligned} \underset{\boldsymbol{w} \in \mathcal{H}}{\text{minimize}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{s.t} \quad & y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i) \rangle + b) \geq 1, \ \forall i. \end{aligned} \tag{2.9}$$

This is equivalent to a 0-1 loss function with ℓ_2-norm regularization of the weights.

So far the data, $\mathcal{D}$, has been assumed linearly separable in $\mathcal{H}$. However, in most real life problems the data are contaminated with noise and this assumption far from holds. In order to avoid overfitting, a tradeoff between the empirical risk and the complexity can be achieved by introducing slack variables, ξ_i, in Equation (2.8). This essentially relaxes the hard margin for the non-separable case, by allowing for some error in the training set. [CV95]

$$y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i, \text{ s.t. } \xi \geq 0, \ \forall i.$$

If $1 \geq \xi_i > 0$ the corresponding observation is correctly classified but lies within the margin, while $\boldsymbol{x}_i$ is misclassified for $\xi_i > 1$. So it follows that $\sum_i \xi_i$ is an upper bound on the number of training errors. The relaxation is illustrated in the right panel of Figure 2.3. The soft margin SVM can be formally formulated by augmenting the objective in Equation (2.9) with a penalty on the errors. [Bur98]

$$\begin{aligned} \underset{\boldsymbol{w} \in \mathcal{H}}{\text{minimize}} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N} \xi_i \\ \text{s.t} \quad & y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \ \forall i. \end{aligned} \tag{2.10}$$

This is still a convex problem, where $C > 0$ controls the tradeoff between the empirical error and the complexity. A larger C penalizes errors harder. The

above formulation is essentially equivalent to using a hinge-loss function with ℓ_2-norm regularization of the weights. [STC04]

The problem in Equation (2.10) can be solved using Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$ giving rise to the following primal optimization problem

$$L_P(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) - 1 + \xi_i) - \sum_{i=0}^{N}\beta_i\xi_i, \quad (2.11)$$

where $\boldsymbol{\theta} = \{\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}, C, \boldsymbol{\xi}\}$.

Since the SVM problem is convex with linear constraints the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for $\boldsymbol{w}$, b, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ to be a solution [Fle81]. Thus solving the SVM problem is equivalent to finding a solution to the KKT conditions. The stationary KKT conditions are given by [Fle81, Bur98]

$$\frac{\mathrm{d}L_P(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N}\alpha_i y_i \varphi(\boldsymbol{x}_i) = \mathbf{0} \Rightarrow \boldsymbol{w} = \alpha_i y_i \varphi(\boldsymbol{x}_i) \quad (2.12)$$

$$\frac{\mathrm{d}L_P(\boldsymbol{\theta})}{\mathrm{d}b} = \sum_{i=1}^{N}\alpha_i y_i = 0 \quad (2.13)$$

$$\frac{\mathrm{d}L_P(\boldsymbol{\theta})}{\mathrm{d}\xi_i} = C - \alpha_i - \beta_i = 0 \Rightarrow C = \alpha_i + \beta_i \quad (2.14)$$

$$\begin{aligned} y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) - 1 + \xi_i &\geq 0 \\ \xi_i &\geq 0 \\ \alpha_i &\geq 0 \\ \beta_i &\geq 0 \\ \alpha_i\big(y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) - 1 + \xi_i\big) &= 0 \\ \beta_i\xi_i &= 0. \end{aligned}$$

From where it follows that

$$\begin{aligned} \alpha_i = 0 &\Rightarrow \xi_i = 0 \text{ and } y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) \geq 1 \\ 0 < \alpha_i < C &\Rightarrow \xi_i = 0 \text{ and } y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) = 1 \\ \alpha_i = C &\Rightarrow \xi_i \geq 0 \text{ and } y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) \leq 1. \end{aligned}$$

These relations reveal one of the important properties of the SVM, namely that, the solution is sparse in $\boldsymbol{\alpha}$. In particular, it is evident that only those training points which is either on or within the margin have a corresponding nonzero α_i. Such training points are called Support Vectors (SVs), and it follows

from Equation (2.12) that the solution and hence the separating hyperplane is spanned by these observations.

Resubstituting Equation (2.12)-(2.14) into the primal in Equation (2.11) and applying the kernel trick leads to the following Wolfe dual formulation where $\boldsymbol{w}$, b and $\boldsymbol{\xi}$ have been eliminated [CV95]

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{maximize}} \quad & L_D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{N}\alpha_i \\ \text{s.t} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{N}\alpha_i y_i = 0, \ \forall i. \end{aligned}$$

Support vector training has thus been reduced to solving the above quadratic programming problem, which can be done using standard techniques. The most common approach is the Sequential Minimal Optimization (SMO) algorithm [Pla98] that breaks the optimization problem into smaller two-dimensional subproblems which can be solved analytically.

For prediction, the decision function in Equation (2.7) can be reformulated using Equation (2.12)

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \operatorname{sgn}\Big(\sum_{i=1}^{N}\alpha_i y_i k(\boldsymbol{x}_i,\boldsymbol{x}) + b\Big).$$

While the bias, b, does not follow explicitly from the training phase, it can be determined using the KKT complementarity conditions. That is, for all SVs with $0 < \alpha_i < C$, the slack variable, ξ_i, is known to be zero and $y_i(\langle \boldsymbol{w}, \varphi(\boldsymbol{x}_i)\rangle + b) = 1$. Thus, by averaging over all SVs with $0 < \alpha_i < C$ a robust estimate of b can be found. For more details on Lagrange multipliers, KKT conditions or a more thorough review of the SVM the reader is referred to, e.g., [Bur98, SS01, STC04].

The basic SVM formulation as presented above has been extended in various ways. For example, Support Vector Regression have been proposed for problems where $y \in \mathbb{R}$ [DBK+97, SS04], and several approaches for multiclass SVMs have been developed, confer with, e.g., [HL02, DK05]. The main drawbacks of the SVM include difficult interpretation of the model weights and defective learning from imbalanced data in which one class heavily outnumbers the other. The latter is a general problem (see, e.g., [Wei04] for a review). Common approaches to improve the SVM performance in imbalanced problems include oversampling the minority class, undersampling the majority class or a combination of the two [CBHK02, RK04, CCB09, TZCK09]. The underlying mechanism for the

lack of performance caused by class imbalance is discussed in, e.g., [WC03, AKJ04]. However, class imbalance also effects the more general problem of variance inflation in small sample high-dimensional problems which is addressed in [AH12, AH13] and is the topic of Chapter 4 of this thesis.

Contributions of this thesis

The performance of the SVM depends on the choice of the SVM margin parameter, C, and the kernel hyperparameter(s). The most common approach to parameter selection is to perform an exhaustive grid search over a predefined range of parameters, and then choose the best setting by minimizing the Cross-Validation (CV) error. However, this approach easily becomes computational infeasible for large-scale problems. In [HAH11] (see Appendix D) this challenge is addressed for large-scale multiclass problems. Several attempts at reducing the complexity of finding the optimal combination of parameters have been made for both binary classification [Joa01, Wah99, VC00] and multiclass problems [LdC08, VM11, VHB10]. However, as argued in [DKP03], all of these approximations are inferior to 5-fold CV. Inspired by previous studies on binary problems suggesting that the intercluster distance in the RKHS is correlated with the optimal kernel hyperparameter [WW09, XXCJ10], [HAH11] proposes a novel algorithm for selecting the intrinsic parameters of the SVM based on fitting a Minimum Enclosing Ball (MEB) to the class means in $\mathcal{H}$. That is, the optimal kernel hyperparameter is chosen by maximizing the size of the MEB. By exploiting the sublinear algorithm for fitting a MEB in a finite-dimensional space derived in [CHW10], randomized approximations for fitting the MEB in the infinite-dimensional feature space are derived having highly competitive time complexities . The new heuristic is compared to standard distance measures like the mean, median, maximum, and minimum on images from the ALOI data set [GBS05]. It is found that only the proposed MEB approach peaks at the optimal kernel hyperparameter as defined by 5-fold CV.

Chapter 3

Denoising by Kernel PCA

The main challenge in denoising by kernel Principal Component Analysis is the inverse problem consisting of finding a point in input space that corresponds to the denoised projection in feature space. This is also known as the pre-image problem, to which this chapter is devoted. Due to the properties of the mapping, the pre-image problem is inherently ill-posed for many choices of kernel function.

Section 3.1 opens with a short description of the pre-image problem, which is followed by an overview of the existing algorithms for reconstructing the pre-image in Section 3.2. The chapter is concluded with a summary of this thesis' contributions in relation to improved pre-image estimation.

3.1 The Pre-Image Problem

Kernel Principal Component Analysis (PCA) denoising can be though of in three steps. First, the noise contaminated input observation, $\boldsymbol{x}$, is mapped to the kernel induced feature space. Secondly, the image, $\varphi(\boldsymbol{x}) \in \mathcal{H}$, is projected onto a q-dimensional principal subspace giving $P_q\varphi(\boldsymbol{x})$. Finally, this projection should be mapped back into input space and a new and hopefully less noisy point $\boldsymbol{z} = \varphi^{-1}(P_q\varphi(\boldsymbol{x}))$ is obtained. $\boldsymbol{z} \in \mathcal{X}$ is then called the pre-image of $P_q\varphi(\boldsymbol{x})$. The last step constitutes the main challenge in stable denoising using kernel PCA and several of the papers in this thesis relates to approximating the inverse mapping, φ^{-1}.

For many nonlinear kernels the dimensionality of the feature space is much larger than the dimensionality of the original input space, and therefore φ cannot be onto (surjective). Furthermore, whether φ is one-to-one (injective) depends on the chosen kernel function. Hence, often φ will not be invertible and the pre-image problem is inherently ill-posed. As illustrated in Figure 3.1 not all points in $\mathcal{H}$ or even in the span of the φ-mapped training data is the image of any point in $\mathcal{X}$, finding the pre-image is not well-defined and instead of seeking an exact pre-image most approaches relaxes the quest to that of finding a good approximation. [SMB+99]

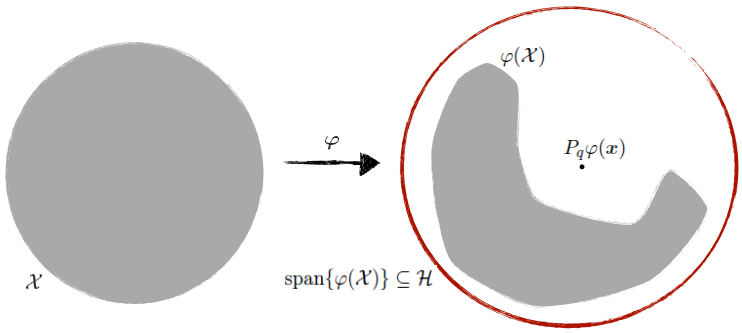


Figure 3.1: Illustration of the pre-image problem. Not all points in $\mathcal{H}$ or even the span of the φ-mapped training points are the image of any point in $\mathcal{X}$. Thus, even for a point given as a linear combination of the mapped training data (e.g., a kernel PCA projection), finding the pre-image is not well-defined. Modified from [SMB+99].

3.2 Pre-image Estimation Schemes

The following section opens with an overview of related work on pre-image estimation before the contributions of this thesis are made explicit at the end of the section. The problem of finding a stable pre-image estimator has been addressed in a variety of ways. The proposed methods can roughly be divided into three groups based on which optimality criteria is used for the approximation.

The first group includes the original work by [SSM98, MSS$^+$99] and consist of methods which estimate the pre-image by minimizing the feature space distance, i.e., the pre-image is sought as a point in input space that maps as close as possible to the feature space projection

$$\boldsymbol{z} = \underset{\boldsymbol{z}\in\mathcal{X}}{\operatorname{argmin}} \; ||\varphi(\boldsymbol{z}) - P_q\varphi(\boldsymbol{x})||^2. \tag{3.1}$$

The idea is illustrated in Figure 3.2. This problem is inherently nonlinear and nonconvex for many choices of kernel function, making it entirely nontrivial to find a reliable pre-image.

The cost function in Equation (3.1) can be expanded to

$$\begin{aligned} ||\varphi(\boldsymbol{z}) - P_q\varphi(\boldsymbol{x})||^2 &= \langle\varphi(\boldsymbol{z}), \varphi(\boldsymbol{z})\rangle + \langle P_q\varphi(\boldsymbol{x}), P_q\varphi(\boldsymbol{x})\rangle - 2\langle\varphi(\boldsymbol{z}), P_q\varphi(\boldsymbol{x})\rangle \\ &= k(\boldsymbol{z},\boldsymbol{z}) - 2\sum_{n=1}^{N}\xi k(\boldsymbol{z},\boldsymbol{x}_n) + \Omega, \end{aligned} \tag{3.2}$$

where the last equality follows from collecting all the $\boldsymbol{z}$-independent terms (originating from $\langle P_q\varphi(\boldsymbol{x}), P_q\varphi(\boldsymbol{x})\rangle$) in Ω, and letting $\xi = \tilde{\xi}_n + \frac{1}{N}(1 - \sum_{j=1}^{N}\tilde{\xi}_j)$, where $\tilde{\xi}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$ as defined in Equation (2.5). Please confer with [Abr09] for full derivations.

For many choices of kernel, and in particular all Radial Basis Function (RBF) kernels, $k(\boldsymbol{x},\boldsymbol{x})$ is constant, hence Equation (3.1) is equivalent to

$$\boldsymbol{z} = \underset{\boldsymbol{z}\in\mathcal{X}}{\operatorname{argmin}} \; -\sum_{n=1}^{N}\xi k(\boldsymbol{z},\boldsymbol{x}_n). \tag{3.3}$$

The original work in [MSS$^+$99] focused on the Gaussian kernel and derived a fixed-point iteration by setting the gradient of Equation (3.3) to zero, thereby arriving at

$$\boldsymbol{z}_{t+1} = \frac{\sum_{n=1}^{N}\xi_n \exp(-\gamma||\boldsymbol{z}_t - \boldsymbol{x}_n||^2)\boldsymbol{x}_n}{\sum_{n=1}^{N}\xi_n \exp(-\gamma||\boldsymbol{z}_t - \boldsymbol{x}_n||^2)}. \tag{3.4}$$

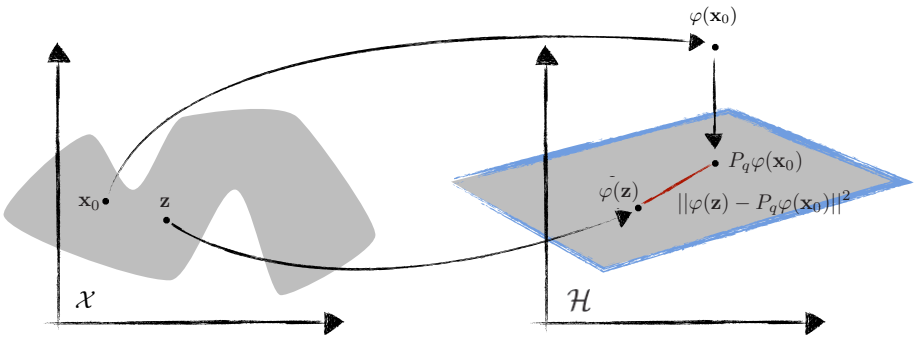


Figure 3.2: Kernel PCA denoising involves mapping the 'noisy' $\boldsymbol{x}_0$ to the kernel induced RKHS and then performing linear PCA in this new space. The crucial part of the denoising process concerns estimating a 'clean' $\boldsymbol{z}$ from the projection in feature space, $\mathcal{H}$. This step is known as the pre-image problem and is inherently ill-posed. Modified from [AH11b].

The approach is easily extended to other kernels as long as the derivative of Equation (3.2) has a closed-form expression. Various suggestions for choosing proper initialization in order to stabilize the iterative approach have been made [TK02, KFS05, TTSL08]. Additionally, [RDT06] proposed to include a preprocessing step where the test datum is projected onto the subspace spanned by the training data in $\mathcal{X}$, thereby arriving at a noniterative scheme for minimizing the distance criteria.

The second set of methods do not seek to minimize Equation (3.2), but rather exploit the fundamental idea of Multidimensional Scaling (MDS) [CC00], where the data are embedded in a lower-dimensional space that preserves the pairwise distances. [KT04] exploited this concept to derive a noniterative algorithm for estimating the pre-image by seeking to preserve the pairwise Euclidean distances in input space and the RKHS. This method does not require the kernel function to be differentiable. Similar to Locally Linear Embedding (LLE), [KT04] only considers a fixed number of neighbors in $\mathcal{H}$ for the reconstruction of the pre-image.

The third group of pre-image approximation methods aim directly at learning the inverse map from $\mathcal{H}$ to $\mathcal{X}$. This approach was initially proposed in [BWS04], where kernel Ridge Regression was used to learn a map between $\mathcal{H}$ and $\mathcal{X}$ from the training data, while [HR09] suggested a similar approach for learning a linear transformation. The modeling of the pre-image map was further elaborated on by [ZL06] where neighborhood information was included and in [ZLY10] where

a penalized methodology was presented.

Other extensions of the above mentioned methods include [ARS07], which exploits the connection between the Nyström formulation of the out-of-sample extension [BPV+04] and the pre-image approximation. While both [NlTF08] and [ZLY06] used regularization in $\mathcal{H}$ either on the projection in feature space or by a weakly supervised prior that puts more weight on positive training samples. Finally, [BZT04] considered the pre-image problem for undirected graphs, and suggested a scheme for reconstructing graphs from the RKHS representation.

In summary, it should be noted that all the pre-image estimators can be formulated as some linear combination of either a subset of or the full training data set, but with different frameworks for learning the weights. Thus, the pre-image is always confined to the span of the training points. A thorough review of the pre-image problem and current estimation schemes can be found in [Abr09, HR11].

Contributions of this thesis

As investigated in detail in [AH11b] the orginal fixed-point iteration in Equation (3.4) as proposed by [MSS+99] is sensitive to initialization and suffer from convergence to local minima in particular in the very nonlinear regime. This realization led to the regularized methods suggested in [AH11b, AH11c] (see Appendix A and B). In order to improve stability, Tikhonov regularization is introduced by augmenting the cost function in Equation (3.3) with an ℓ_2-norm penalty on the distance between the pre-image reconstruction and the noisy observation in input space, denoted by $\boldsymbol{x}_0$

$$\rho_{\ell_2}(\boldsymbol{z}) = -\sum_{n=1}^{N} \xi_n k(\boldsymbol{z}, \boldsymbol{x}_n) + \lambda ||\boldsymbol{z} - \boldsymbol{x}_0||^2.$$

Since the penalty term is differentiable, a new fixed-point iteration can easily be derived

$$\boldsymbol{z}_{t+1} = \frac{\gamma \sum_{n=1}^{N} \xi_n \exp\left(-\gamma ||\boldsymbol{z}_t - \boldsymbol{x}_n||^2\right) \boldsymbol{x}_n + \lambda \boldsymbol{x}_0}{\gamma \sum_{n=1}^{N} \xi_n \exp\left(-\gamma ||\boldsymbol{z}_t - \boldsymbol{x}_n||^2\right) + \lambda}.$$

Furthermore, Least Absolute Shrinkage and Selection Operator (LASSO) regularization [Tib96] was suggested for sparse denoising problems by introducing ℓ_1-norm penalty on the reconstruction

$$\rho_{\ell_1}(\boldsymbol{z}) = -\sum_{n=1}^{N} \xi_n k(\boldsymbol{z}, \boldsymbol{x}_n) + \lambda \sum_{j=1}^{D} |z_j|. \tag{3.5}$$

This can be interpreted as a Maximum A Posteriori (MAP) estimate with individual Laplacian priors on the pre-image features z_j. In Equation (3.5), the penalty term is not differentiable and a fixed-point iteration can not be derived. Instead the Generalized Path Seeking (GPS) framework introduced by [Fri08] is used to approximate the path of decreasing sparsity and thereby finding a solution to Equation (3.5). For additional background on smoothing by ℓ_p-norms please refer to, e.g., [RW64].

In [AH11b] a subset of the USPS data base of handwritten digits [Hul94] is used to demonstrate how introducing input space distance regularization both stabilizes the pre-image as measured by the variability between test points and reduces the sensitivity to initialization. Additionally, ℓ_1-norm regularization is found to significantly improve the reconstruction in terms of visual quality, albeit this is achieved with a sacrifice in the Mean Squared Error (MSE) on this data set. The applicability of the sparse reconstruction is investigated further in [AH11c], where single slice functional Magnetic Resonance Imaging (fMRI) data from a visual block activation experiment is used to demonstrate the potential of sparse denoising using kernel PCA. The performance is reported using a combined prediction/reproducibility metric, where the predictive power is measured as the accuracy of a linear discriminant in the q-dimensional principal subspace while the reproducibility is quantified using the split-half resampling procedure, Nonparametric, Prediction, Activation, Influence, Reproducibility, re-Sampling (NPAIRS) as introduced by [SAH$^+$02]. The experiments validates how the sparse pre-image estimate is superior to the dense reconstruction and that the denoised brain maps are not only visual appealing but also highly reproducible.

While linear methods are well trusted in the neuroimaging community the practical use of nonlinear kernels have been limited. The work contained in [RAMH12] (see Appendix C) is dedicated to the use of kernel PCA denoising in neuroimaging. First, kernel PCA denoising is implemented as part of the image preprocessing pipeline, and secondly, the hyperplane navigation procedure for linear models [STC$^+$08] is extended to navigate nonlinear manifolds by using pre-image estimation to generate brain maps in the continuum between experimentally defined classes. Two fMRI data sets recorded under finger movement [RHM$^+$12] and visual stimuli [HGF$^+$01], respectively, are used to evaluate the model. The performance is measured by jointly evaluating the prediction accuracy and the pattern reproducibility of the denoised brain scans within the NPAIRS framework. The experiments work as a proof of concept study indicating that including nonlinear kernel PCA denoising in the preprocessing pipeline leads to relatively large gains in the reproducibility with no sacrifice in terms of predictive power.

The last part of this thesis regarding the pre-image problem involves the semisu-

pervised framework devised in [HAH13] (see Appendix D). Additional to the iterative scheme for finding principal components biased toward the available class information as mentioned in Section 2.2, [HAH13] also introduces an alternative approach for incorporating label information in kernel PCA denoising. This is done by deriving a fixed-point iteration for estimating the pre-image using the graph kernel function, k_g, introduced by [SNB05]

$$k_g(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_{\boldsymbol{x}}^\top (\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})^{-1} \boldsymbol{L} \boldsymbol{k}_{\boldsymbol{x}'} = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_x^\top \boldsymbol{M} \boldsymbol{k}_{\boldsymbol{x}'},$$

where $\boldsymbol{k}_{\boldsymbol{x}'} = [k(\boldsymbol{x}_1, \boldsymbol{x}'), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}')]^\top$ and $\boldsymbol{L}$ is the combinatorial graph Laplacian, defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{K}$, where $\boldsymbol{D}$ is a diagonal matrix with $D_{ii} = \sum_{i=1}^N K_{ij}$. The last equality follows form letting $\boldsymbol{M} = (\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})^{-1}\boldsymbol{L}$. This graph-based kernel warps the corresponding RKHS to account for the manifold structure imposed by both labeled and unlabeled data points.

By setting the derivative of the cost function in Equation (3.2) to zero for this new kernel, a fixed-point iteration can be found

$$\begin{aligned} \boldsymbol{z}_{t+1} = &\frac{\left[(\boldsymbol{M} \circ (\boldsymbol{k}_{\boldsymbol{z}_t} \boldsymbol{k}_{\boldsymbol{z}_t}^\top - \boldsymbol{k}_{\boldsymbol{z}_t} (\boldsymbol{K}\boldsymbol{\xi})^\top - (\boldsymbol{K}\boldsymbol{\xi}) \boldsymbol{k}_{\boldsymbol{z}_t}^\top))\mathbf{1}\right]^\top \boldsymbol{X}}{(\boldsymbol{k}_{\boldsymbol{z}_t}^\top \boldsymbol{M} + \boldsymbol{\xi}^\top - 2\boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{M}) \boldsymbol{k}_{\boldsymbol{z}_t}} \\ &+ \frac{\left[\boldsymbol{M} \circ \boldsymbol{\xi} \circ \boldsymbol{k}_{\boldsymbol{z}_t}\right]^\top \boldsymbol{X}}{(\boldsymbol{k}_{\boldsymbol{z}_t}^\top \boldsymbol{M} + \boldsymbol{\xi}^\top - 2\boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{M}) \boldsymbol{k}_{\boldsymbol{z}_t}}. \end{aligned}$$

For simplicity it is assumed that the pre-image itself is not part of $\boldsymbol{K}$. Thereby the inversion of $(\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})$ which scales cubically can be avoided at every iteration. The effect of this assumption is minimal when the manifold is well defined by the training data. Both synthetic data and images from the Amsterdam Library of Object Images (ALOI) [GBS05] are used to experimentally validate that kernel PCA denoising is improved when using the graph kernel as opposed to the classical Gaussian kernel. Furthermore, the combination of the graph kernel and the semisupervised kernel PCA is found to be superior to the other schemes.

The regularized approaches have subsequently been applied within other domains. In [LP11, LP12b, LP13] the ℓ_2-norm regularized approach is used for speech enhancement. Together with the method of [KT04], input space distance regularization is found to outperform the estimators proposed by [MSS+99] and [HR09]. [LP12c] successfully applies kernel PCA denoising with the ℓ_2-norm penalized pre-image estimator to musical noise suppression, and finally, the regularized kernel PCA denoising approach is used for face detection as part of a system for improved face recognition in [ACS+11]. Furthermore, [KHR+13] uses the ℓ_2-norm regularized method as a benchmark when evaluating their proposed pre-image estimator. Additionally, the papers related to regularized pre-image estimation have been cited by [CLI+11, HR11, Nie11, CSL+12, LP12a,

RAMH12, TFX12][1], while the paper on kernel PCA denoising in neuroimaging has been cited by [SCB13, FWXT13].

[1]The publications mentioned include citations of [AH09], which is not included in this thesis

Chapter 4

Variance Inflation

The final chapter of the summary report focuses on the variance inflation problem inherent in small sample high-dimensional learning.

Section 4.1 introduces variance inflation by focusing on the lack of generalizability in high-dimensional linear models, which is followed by a outline of the contributions of this thesis. The outline is comprised of an exposition of the variance inflation problem in kernel methods and the proposed schemes for restoring the performance of both kernel Principal Components Analysis and the Support Vector Machine.

4.1 Small sample high-dimensional problems

In early years most research focused on small D large N problems and theory for the limiting case as $N \to \infty$, such as the law of large numbers and the central limit theorem, were developed. In general, it was considered good practice to require that the number of samples should be some factor larger than the number of parameters to be estimated. For example, work like [H$^+$70, Hub] suggested $N/D \geq 5$. However, the recent development within data acquisition and computing technologies have lead to a new challenge where the dimensionality of the data may be much larger than the number of samples, i.e., $D \gg N$. Examples include image analysis, microarray analysis, document classification etc.. [JT09]

When the dimensionality is large relative to the number of samples, the data become sparse and possibly insufficient to describe significant relations. These kind of problems go under a variety of names such as small-sample high-dimensional problems, large D (or p) small N problems, ill-posed problems or even underdetermined systems. The curse of dimensionality as introduced in [BBB66], is often used as a generic term for the challenges imposed by working in such high-dimensional spaces.

A classical way of dealing with the curse of dimensionality is to find a lower-dimensional representation of the data where the signal manifold is preserved. One way of doing this is by Principal Component Analysis (PCA). However, it has been shown that extracting such principal components from small samples in high dimensions is not straightforward in cases where the intrinsic dimension of the signal manifold exceeds the sample size. Thus, even though linear PCA is well understood and conceptually simple, learning the Principal Components (PCs) in high-dimensional spaces becomes highly nontrivial as argued in publications like [KHS$^+$01b, HR07].

The lack of data relative to the complexity of the problem also leads the empirical covariance matrix to become rank-deficient. This not only affects PCA but impair most conventional statistical methods, for example, linear regression, fail as the Maximum Likelihood (ML) estimate of the parameter vector can not be uniquely estimated if $D \leq N$, since $\boldsymbol{X}^\top \boldsymbol{X}$ will be singular. A classical way of dealing with this ambiguity is to introduce regularization leading to penalized least squares or penalized maximum likelihood methods like ridge regression [HK70] and the Least Absolute Shrinkage and Selection Operator (LASSO) [Tib96]. Another effect present in high-dimensional spaces is that the Euclidean distance measure is known to deteriorate as all pairwise distances become almost equal. This cause any nearest neighbor method to become meaningless, an effect also known as distance concentration which has among others been

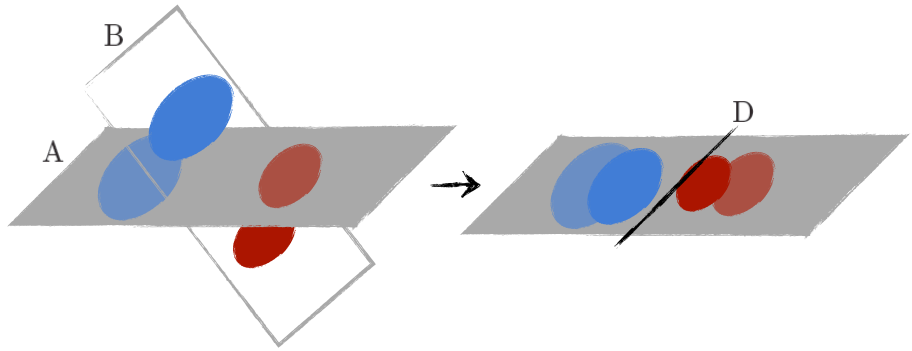


Figure 4.1: Illustration of the variance inflation problem in PCA. Because PCA maximizes variance, small data sets in high dimensions will be overfitted. When the PCA subspace (A) is applied to a test data set (B) the projected data will have smaller variance. This leads to lack of generalizability if the training data are used to train a classifier, say a linear discriminant (D). The effect is pronounced if the problem is imbalanced. Modified from [AH11a].

explored by [BGRS99, HAK+00, AY01, FWV07].

Thus, it is evident that estimation becomes nontrivial when $D \gg N$. Since the training data can maximally span a N-dimensional subspace of the full data space any contribution orthogonal to the learned subspace will be missed when generalizing to new unseen data. As shown in [KHS+01b, AH11a] for linear PCA, the lost orthogonal projection can manifest as the test data following different probability laws with smaller variance than the training data. This effect will also be referred to as variance inflation because the variance estimate on the training data is defectively large as is illustrated in Figure 4.1.

In [KHS+01b] it was shown that there is a sharp transition from a condition where the amount of data is insufficient and no learning is possible to a regime where the learned PCs become increasingly more accurate as the sample size increase. For sample sizes below the phase transition point the eigenvectors are completely random as there is no learning at all. Then as the sample size increase, initially the first PC stabilizes, then the second and so forth. In the regime where learning is possible but still incomplete, the effect can be interpreted as overfitting of the PCs to the training data. [KHS+01b] proposes a Leave-One-Out (LOO) framework for renormalizing the test projections and thereby restore the generalizability of linear PCA. Since the projections in linear PCA by definition are uncorrelated it is possible to renormalize the components independently. When assuming normality of the projections an affine transfor-

mation can be used to adjust for the variance inflation. The scale factor, which is determined in the LOO scheme, is simply given by the ratio of the standard deviation of the training and test set projections, respectively.

Contributions of this thesis

Even though the main focus of [AH11a] (see Appendix E) is the variance inflation problem in kernel PCA, a simpler approximation to the LOO scheme in [KHS+01b] for renormalizing the linear PCA projections is also proposed. The full LOO procedure requires the computation of N Singular Value Decompositions (SVDs) of a $(N-1) \times (N-1)$ matrix, and so the time complexity becomes $\mathcal{O}(N^4)$ [GVL96]. A significant speed-up can be achieved by assuming that the changes in the eigenvectors of the covariance matrix when reducing the sample from N to $N-1$ are small, and thus the eigenvectors of the full sample can be used as an approximation. This simplification effectively reduces the time complexity to $\mathcal{O}(N^3)$. As derived in [AH11a], an error of order $\mathcal{O}(\frac{1}{N})$ is introduced in the value of the eigenvalues and possible perturbation of the eigenvectors.

Moving to the issues of variance inflation in kernel embeddings, the publications [AH11a, AH12, AH13] (see Appendix E, F, and G) addresses how the challenges of small samples in high-dimensions extends from linear models to kernel learning exemplified by kernel PCA and the Support Vector Machine (SVM). An issue which has only recently started to receive proper attention.

The statistical properties of kernel PCA have been studied in depth [BBZ07, HR04a, HR04b, STW03, ZB06]. Nevertheless the extremely ill-posed case has not been assessed until [AH11a]. For the Gaussian kernel, kernel PCA approaches linear PCA as $\gamma \to 0$, and as discussed in the previous section variance inflation can be expected to deteriorate the result. In [AH11a] it was shown that the same effect can be expected in the nonlinear regime as γ increases.

By splitting a test datum, $\boldsymbol{x}$, in the orthogonal and the parallel components with respect to the subspace spanned by the training data ($\boldsymbol{x} = \boldsymbol{x}^{\perp} + \boldsymbol{x}^{\parallel}$), it can be realized that the test projection (see Equation (2.4)) acquires a common scaling factor of $\exp(-\gamma||\boldsymbol{x}^{\perp}||^2)$ due to the lost orthogonal projection

$$\begin{aligned}\beta_i &= \sum_{n=1}^{N} \alpha_{in} \tilde{k}(\boldsymbol{x}, \boldsymbol{x}_n) = \sum_{n=1}^{N} \alpha_{in} \exp\big(-\gamma(||\boldsymbol{x}^{\parallel} - \boldsymbol{x}_n||^2 + ||\boldsymbol{x}^{\perp}||^2)\big) \\ &= \exp(-\gamma||\boldsymbol{x}^{\perp}||^2) \sum_{n=1}^{N} \alpha_{in} k(\boldsymbol{x}^{\parallel}, \boldsymbol{x}_n), \end{aligned} \tag{4.1}$$

which can be arbitrarily small in the nonlinear regime (large γ).

[AH11a] proposes a coordinate-wise nonparametric renormalization procedure for restoring the generalizability in kernel PCA. Two new challenges are faced when introducing renormalization in the kernelized version of PCA as opposed to the linear counterpart, namely that the components may not be uncorrelated and the projections can potentially by strongly non-normal. Therefore, [AH11a] suggests to check for dependencies through a pairwise permutation test of the mutual information [Mod89] before coordinate-wise renormalization is employed. The potentially non-normal distribution of the projections is overcome through the nonparametric generalization of the affine scaling method suggested in [KHS+01b]. If the training and test sets have the same prior distribution a nonparametric procedure can be derived by using standard histogram equalization to restore the variation in the test set. The problem of calibrating for an unknown monotonic transformation is a common task in image processing equivalent of equalizing two equal sized images [GW77]. The approach is illustrated in Figure 4.2. This renormalization procedure only requires two additional $\mathcal{O}(N \log(N))$[1] operations for sorting the values of the training and test sets.

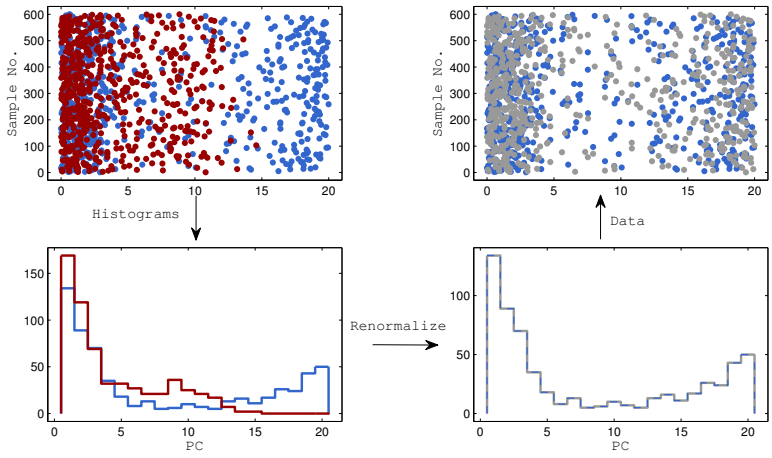


Figure 4.2: Illustration of variance inflation and the histogram equalization approach to restoring generalizability. The equalization is done idenpendently for each PC. The training data are marked with blue, the test data are red, while the renormalized test data are gray. Modified from [AH11a]

[1] The average time complexity of quicksort

The viability of the proposed nonparametric renormalization scheme is demonstrated on the USPS data base of handwritten digits [Hul94] and functional Magnetic Resonance Imaging (fMRI) brain state decoding. The proposed scheme is evaluated by training a linear discriminant in the principal subspace. Compared to conventional kernel PCA, including the renormalization step is found to lower the Mean Squared Error (MSE) significantly in both experiments.

The problem of restoring generalizability in supervised learning is generally less well understood. In [AH12, AH13] it is investigated how the variance inflation problem extends from kernel PCA to supervised kernel learning including the case of the SVM. While variance inflation is expected in ill-posed supervised problems in general, it may be further amplified in imbalanced data. In the following the signed value of the decision function of the SVM given in Equation (2.3) will be referred to as the decision value, i.e., $\sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b$.

Similar to Equation (4.1), the lost orthogonal projection, caused by the mismatch between the subspaces spanned by the training and test data, respectively, is found to lead to a common scaling factor of the decision values. Analogue to the kernel PCA problem, the scaling factor is determined by splitting the test datum in its orthogonal and parallel components

$$\begin{aligned}\sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b &= \sum_{i=1}^{N} \alpha_i y_i \exp\big(-\gamma(||\boldsymbol{x}^{\parallel} - \boldsymbol{x}_i||^2 + ||\boldsymbol{x}^{\perp}||^2)\big) + b \\ &= \exp(-\gamma||\boldsymbol{x}^{\perp}||^2) \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}^{\parallel}, \boldsymbol{x}_i) + b.\end{aligned}$$

As introduced in [AH12] a nonparametric scheme for histogram equalization, similar to that suggested for kernel PCA, can be applied to the decision values in order to adjust for the variance inflation. By proposing a LOO scheme for learning the transfer function, [AH13] extends the renormalization framework to address problems where the training and test sets do not follow the same prior class distribution. However, this generalization is achieved at the expense of a significant increase in the computational burden, since this LOO scheme requires the training of N separate SVMs.

The variance inflation problem and the impact of the proposed renormalization scheme in the SVM are illustrated on a wide variety of benchmark machine learning data sets from the University of California, Irvine machine learning repository (UCI) [FA10] and the Kent Ridge biomedical data set repository (KR)[2]. The performance is reported in terms of both the accuracy and the

[2]Available at `http://datam.i2r.a-star.edu.sg/datasets/krbd/`

Geometric mean given by $G = \sqrt{\text{sensitivity} \cdot \text{specificity}}$ [KM97]. In most cases renormalization is found to significantly improve the performance, while the proposed scheme never is found to decidedly impair the classification ability.

The paper on the variance inflation problem in kernel PCA has subsequently been cited by [AGPCVH13, BHH+13, FWXT13, SZD13]. [TMA+12] mentions the variance inflation problem in neuroimaging data in relation to brain-computer interfacing, while [MFA+13] applies the non-parametric renormalization scheme prior to classification of Parkinsonian disorders from fMRI. Finally, [GMARH12] proposes a hold-out procedure for determining the correction factor for variance inflation in linear PCA with a time complexity that is lower than the full LOO approximation and independent of the number of samples.

Chapter 5

Conclusion

The work contained in this thesis have addressed two main challenges. Namely, the instability related to pre-image estimation and the variance inflation problem in kernel learning. In this chapter, the main conclusions are summarized.

Summary

The overall focus of this dissertation has been on improved kernel learning. Moreover, the two main areas of research have been denoising by kernel Principal Component Analysis (PCA) and the variance inflation problem in kernel methods.

Stable pre-image estimation is vital to meaningful denoising by kernel PCA. The pre-image problem is inherently ill-posed and some of the most commonly used estimation schemes suffer from instability in the nonlinear regime. In this thesis it was shown how current estimation schemes demonstrate high sensitivity to initialization and suffer from convergence to local minima, causing the reconstruction to exhibit large variability. The current work proposed two approaches for stabilizing the pre-image estimation. By augmenting the existing cost function with an ℓ_2-norm penalty term, a more robust pre-image reconstruction was derived, while ℓ_1-norm regularization was introduced for sparse problems. Furthermore, frameworks for exploiting label information to improve the denoising ability in semisupervised problems were derived.

Experiments on handwritten digits showed that ℓ_2-norm regularization provides a more stable pre-image estimate with little to no sacrifice in terms of denoising ability measured by the Mean Squared Error (MSE). Furthermore, it was experimentally validated that the regularized scheme reduces the variability between test points, reduces the sensitivity to initialization, and finally provides a better visual result. The ℓ_1-norm penalized reconstruction was applied to data ranging from handwritten digits to neuroimaging data in form of functional Magnetic Resonance Imaging (fMRI). Sparse pre-image estimation was found to improve visual quality and reproducibility, albeit an increase in the MSE was induced in the handwritten digits. Finally, extensive studies of the applicability of nonlinear denoising by kernel PCA and pre-image estimation within neuroimaging was performed. These experiments clearly indicated that nonlinear image denoising leads to significant gains in the reproducibility of the brain maps while maintaining the accuracy. The performance was measured jointly by the predictive power and the reproducibility of the brain maps extracted from classification models within the Nonparametric, Prediction, Activation, Influence, Reproducibility, re-Sampling (NPAIRS) framework.

The challenges induced by variance inflation in small sample high-dimensional problems was addressed for both unsupervised and supervised kernel learning. Analogue to linear methods, kernel models were found to be biased towards the training set. A geometric interpretation of the variance inflation problem was provided, namely that test data loose their orthogonal projection, when the training data are insufficient to describe the signal manifold.

It was demonstrated how dimensionality reduction by both PCA or kernel PCA can be impaired by inflation of the training set variance leading to a lack of generalization in the extremely ill-posed case. Initially, a speed-up for the for the full Leave-One-Out (LOO) renormalization scheme for linear PCA was derived, while both an exact LOO renormalization procedure and a nonparametric approximation at a lower cost was proposed for kernel PCA. The viability of the proposed scheme for kernel PCA was proved experimentally on both handwritten digits and fMRI brain state decoding. It was shown that renormalization restores generalizability as measured by the MSE of a linear discriminant trained in the principal subspace.

It was further shown how the variance inflation problem extends to the Support Vector Machine (SVM), where generalizability was found to deteriorate in ill-posed imbalanced data due to the lost orthogonal projection incurred as the training data fails to span the entire feature space. The renormalization scheme introduced for kernel PCA was adopted to the SVM formulation, leading to a nonparametric scheme for renormalizing the decision values. Viability was proven experimentally on numerous benchmark machine learning data sets as well as fMRI. The results confirmed that the proposed renormalization procedure restores generalizability, as measured by the MSE and the Geometric-mean. In most domains performance was improved while renormalization never deteriorated the result. Furthermore, renormalization was found to reduce the sensitivity to the choice of kernel hyperparameter as well as leading to a more nonlinear optimal kernel embedding in several of the data sets, thereby suggesting that the signal manifold is more nonlinear than anticipated from classical SVM learning.

Apart from the work on kernel PCA denoising and the variance inflation problem, this thesis also introduced a novel heuristic for kernel hyperparameter selection in large scale multi-class problems, which was validated experimentally on image data.

Appendix A

Regularized Pre-image Estimation for Kernel PCA De-noising

Trine Julie Abrahamsen and Lars Kai Hansen. Regularized Pre-image Estimation for Kernel PCA De-noising. *Journal of Signal Processing Systems*, 65:403–412, 2011.
doi:10.1007/s11265-010-0515-4.

Citations
A shorter version of this paper was originally published at *Proceedings of 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2009 under the title Input space regularization stabilizes pre-images for kernel PCA de-noising. The original paper has been cited by [AH10, CLI+11, HR11, LP11, Nie11, CSL+12, LP12a, LP12b, LP12c, RAMH12, LP13]. While this longer version has been cited by [AH10, ACS+11, AH11c, RAMH12, TFX12, KHR+13].

J Sign Process Syst (2011) 65:403–412
DOI 10.1007/s11265-010-0515-4

Regularized Pre-image Estimation for Kernel PCA De-noising

Input Space Regularization and Sparse Reconstruction

Trine Julie Abrahamsen · Lars Kai Hansen

Received: 15 January 2010 / Revised: 26 July 2010 / Accepted: 26 July 2010 / Published online: 11 August 2010

Abstract The main challenge in de-noising by kernel Principal Component Analysis (PCA) is the mapping of de-noised feature space points back into input space, also referred to as "the pre-image problem". Since the feature space mapping is typically not bijective, pre-image estimation is inherently illposed. As a consequence the most widely used estimation schemes lack stability. A common way to stabilize such estimates is by augmenting the cost function by a suitable constraint on the solution values. For de-noising applications we here propose Tikhonov input space distance regularization as a stabilizer for pre-image estimation, or sparse reconstruction by Lasso regularization in cases where the main objective is to improve the visual simplicity. We perform extensive experiments on the USPS digit modeling problem to evaluate the stability of three widely used pre-image estimators. We show that the previous methods lack stability in the is non-linear regime, however, by applying our proposed input space distance regularizer the estimates are stabilized with a limited sacrifice in terms of de-noising efficiency. Furthermore, we show how sparse reconstruction can lead to improved visual quality of the estimated pre-image.

Keywords Kernel PCA · Pre-image · Regularization · De-noising · Sparsity

T. J. Abrahamsen (✉) · L. K. Hansen
DTU Informatics, Technical University of Denmark,
Richard Petersens Plads, building 321,
2800 Kgs. Lyngby, Denmark
e-mail: tjab@imm.dtu.dk

L. K. Hansen
e-mail: lkh@imm.dtu.dk

1 Introduction

We are interested in unsupervised learning methods for de-noising. If necessary we will use non-linear maps to project noisy data onto a clean signal manifold. Kernel PCA and similar methods are widely used candidates for such projection beyond conventional linear unsupervised learning schemes like principal component analysis (PCA), independent component analysis (ICA), and non-negative matrix factorization (NMF). The basic idea is to implement the projection in three steps, in the first step we map the original input space data into a feature space. The second step then consists of using a conventional linear algorithm, like PCA, to identify the signal manifold by linear projection in feature space. Finally, in the third step we estimate the de-noised input space points that best correspond to the projected feature space points. The latter step is referred to as the *pre-image problem*. Unfortunately, finding a reliable pre-image is entirely non-trivial and has given rise to several algorithms [2, 4, 8, 9, 14]. *In this work we experimentally analyze the stability of the estimated pre-images from the most used of these algorithms, we suggest to introduce regularization in order to improve the performance and stability relative to the existing approaches. If the aim is stabilization, Tikhonov input space regularization is recommendable whereas sparse reconstruction by Lasso regularization is found superior for sparse data when the aim is improved visual quality.*

Let us recapitulate some basic aspects of de-noising with kernel PCA. Let $\mathcal{F}$ be the Reproducing Kernel Hilbert Space (RKHS) associated with the kernel function $k(\mathbf{x}, \mathbf{x}') = \varphi(\mathbf{x})^T\varphi(\mathbf{x}')$, where $\varphi : \mathcal{X} \mapsto \mathcal{F}$ is a possibly nonlinear map from the D-dimensional input space

404 J Sign Process Syst (2011) 65:403–412

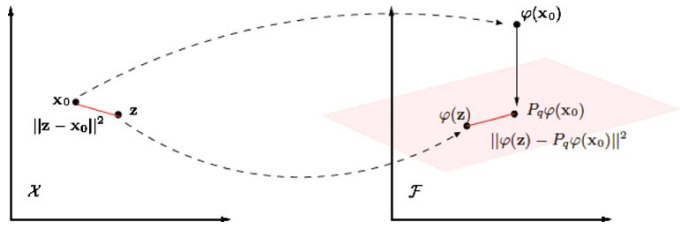


Figure 1 The pre-image problem in kernel PCA de-noising concerns estimating $\mathbf{z}$ from $\mathbf{x}_0$, through the projection of the image onto the principal subspace. Presently available methods for pre-image estimation lead to unstable pre-images because the inverse is ill-posed. We show that simple input space regularization, with a penalty based on the distance $||\mathbf{z} - \mathbf{x}_0||$ leads to a stable pre-image.

$\mathcal{X}$ to the high (possibly infinite) dimensional feature space $\mathcal{F}$ (see notation[1]). In de-noising and a number of other applications it is of interest to reconstruct a data point in input space from a point in feature space. Hence, applying the inverse map of φ. Given a point, Ψ, in feature space the pre-image problem thus consists of finding a point $\mathbf{z} \in \mathcal{X}$ in the input space such that $\varphi(\mathbf{z}) = \Psi$. $\mathbf{z}$ is then called the pre-image of Ψ. For many non-linear kernels $\dim(\mathcal{F}) \gg \dim(\mathcal{X})$ and φ is not surjective. Furthermore, whether φ is injective depends on the choice of kernel function. As a function $f : X \mapsto Y$ has an inverse iff it is bijective, we do not expect φ to have an inverse. When φ is not surjective, it follows that not all points in $\mathcal{F}$ or even the span of $\{\varphi(\mathcal{X})\}$ is the image of some $\mathbf{x} \in \mathcal{X}$. Finally, when φ is not injective, any recovered pre-image might not be unique. Thus the pre-image problem is ill-posed [1, 3, 4, 8, 9, 12, 14]. As we can not expect an exact pre-image, we follow [9] and relax the quest to find an *approximate pre-image*, i.e., a point in input space which maps into a point in feature space 'as close as possible' to Ψ (Fig. 1).

2 Kernel PCA

Kernel Principal Component Analysis is a nonlinear generalization of linear PCA, in which PCA is carried out in the feature space $\mathcal{F}$ mapped data [13]. However, as $\mathcal{F}$ can be infinite dimensional we can not work directly with the feature space covariance matrix. Fortunately, the so-called kernel trick allows us to formulate nonlinear extensions of linear algorithms when these are expressed in terms of inner-products.

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be N training data points in $\mathcal{X}$ and $\{\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_N)\}$ be the corresponding images in $\mathcal{F}$. The mean of the φ-mapped data points is denoted $\bar{\varphi} = \frac{1}{N}\sum_{n=1}^{N}\varphi(\mathbf{x}_n)$ and the 'centered' images are given by $\tilde{\varphi}(\mathbf{x}) = \varphi(\mathbf{x}) - \bar{\varphi}$. Now, let $\boldsymbol{K}$ denote the kernel matrix with element $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, then kernel PCA can be performed by solving the eigenvalue problem

$$\widetilde{\mathbf{K}}\boldsymbol{\alpha}_i = \lambda_i \boldsymbol{\alpha}_i \tag{1}$$

where $\widetilde{\mathbf{K}}$ is the centered kernel matrix defined as $\widetilde{\mathbf{K}} = \mathbf{K} - \frac{1}{N}\mathbf{1}_{NN}\mathbf{K} - \frac{1}{N}\mathbf{K}\mathbf{1}_{NN} + \frac{1}{N^2}\mathbf{1}_{NN}\mathbf{K}\mathbf{1}_{NN}$.

The projection of a φ-mapped test point onto the *i'th* principal component is

$$\beta_i = \tilde{\varphi}(\mathbf{x})^T\mathbf{v}_i = \sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\mathbf{x})^T\tilde{\varphi}(\mathbf{x}_n) = \sum_{n=1}^{N}\alpha_{in}\tilde{k}(\mathbf{x}, \mathbf{x}_n) \tag{2}$$

where $\mathbf{v}_i$ is the *i'th* eigenvector of the feature space covariance matrix and the $\boldsymbol{\alpha}_i$'s have been normalized. The centered kernel function can be found as $\tilde{k}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \frac{1}{N}\mathbf{1}_{1N}\mathbf{k}_{\mathbf{x}} - \frac{1}{N}\mathbf{1}_{1N}\mathbf{k}_{\mathbf{x}'} + \frac{1}{N^2}\mathbf{1}_{1N}\mathbf{K}\mathbf{1}_{N1}$, where $\mathbf{k}_{\mathbf{x}} = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_N)]^T$. The projection of $\varphi(\mathbf{x})$ onto the subspace spanned by the first q eigenvectors will be denoted $P_q\varphi(\mathbf{x})$ and can be found as

$$\begin{aligned} P_q\varphi(\mathbf{x}) &= \sum_{i=1}^{q}\beta_i\mathbf{v}_i + \bar{\varphi} = \sum_{i=1}^{q}\beta_i\sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\mathbf{x}_n) + \bar{\varphi} \\ &= \sum_{n=1}^{N}\tilde{\gamma}_n\tilde{\varphi}(\mathbf{x}_n) + \bar{\varphi} \end{aligned} \tag{3}$$

where $\tilde{\gamma}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$. Kernel PCA satisfies properties similar to those for linear PCA, namely that the squared

[1] Bold uppercase letters denote matrices, bold lowercase letters represent column vectors, and non-bold letters denote scalars. $\mathbf{a}_j$ denotes the *j'th* column of $\mathbf{A}$, while a_{ij} denotes the scalar in the *i'th* row and *j'th* column of $\mathbf{A}$. Finally, $\mathbf{1}_{NN}$ is a $N \times N$ matrix of ones.

J Sign Process Syst (2011) 65:403–412

reconstruction error is minimal and the retained variance is maximal. However, these properties hold in $\mathcal{F}$ not in $\mathcal{X}$. For a more thorough derivation of kernel PCA the reader is referred to, e.g., [13].

3 Approximate Pre-images

Several optimality criteria can be used for the pre-image approximation, see e.g., [1],

$$\text{Distance:} \quad \mathbf{z} = \underset{\mathbf{z}\in\mathcal{X}}{\operatorname{argmin}} \; ||\varphi(\mathbf{z}) - \Psi||^2 \tag{4}$$

$$\text{Co-linearity:} \quad \mathbf{z} = \underset{\mathbf{z}\in\mathcal{X}}{\operatorname{argmax}} \left\langle \frac{\varphi(\mathbf{z})}{||\varphi(\mathbf{z})||}, \frac{\Psi}{||\Psi||} \right\rangle \tag{5}$$

For RBF kernels of the form $k(\mathbf{x}_i, \mathbf{x}_j) = \kappa(||\mathbf{x}_i - \mathbf{x}_j||^2)$ the co-linearity criteria and the distance criteria coincide:

$$\begin{aligned} ||\varphi(\mathbf{z}) - \Psi||^2 &= \langle \varphi(\mathbf{z}), \varphi(\mathbf{z}) \rangle + \langle \Psi, \Psi \rangle - 2 \langle \varphi(\mathbf{z}), \Psi \rangle \\ &= k(\mathbf{z}, \mathbf{z}) + ||\Psi||^2 - 2 \langle \varphi(\mathbf{z}), \Psi \rangle \end{aligned} \tag{6}$$

As $k(\mathbf{z}, \mathbf{z})$ is constant for RBF kernels and $||\Psi||^2$ is independent of $\mathbf{z}$, minimizing $||\varphi(\mathbf{z}) - \Psi||^2$ is equivalent to maximizing the co-linearity. As $\mathcal{F}$ is a RKHS, the distance will be the same before and after centering. However, the expression gets a bit more tedious when using explicit centering as will be shown later.

Thus we seek to minimize the distance between $\varphi(\mathbf{z})$ and Ψ with respect to $\mathbf{z}$. By assuming that Ψ lies in (or close to) the span of $\{\varphi(\mathbf{x}_i)\}$, Ψ can be represented as a linear combination of the training images, i.e. $P_q\varphi(\mathbf{x})$, without loss of generality. When $q = N$ this will translate to projecting Ψ onto the span of $\{\varphi(\mathbf{x}_i)\}$. Thus, we are interested in an expression for

$$\begin{aligned} ||\varphi(\mathbf{z}) - P_q\varphi(\mathbf{x})||^2 = ||\varphi(\mathbf{z})||^2 + ||P_q\varphi(\mathbf{x})||^2 \\ -2\varphi(\mathbf{z})^T P_q\varphi(\mathbf{x}). \end{aligned} \tag{7}$$

The terms will in the following be expanded separately, starting with the first term

$$||\varphi(\mathbf{z})||^2 = \varphi(\mathbf{z})^T\varphi(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) \tag{8}$$

From Eq. 3 and the definition of centering and mean in feature space, we have

$$\begin{aligned} ||P_q\varphi(\mathbf{x})||^2 &= \left(\sum_{i=1}^{q} \beta_i \mathbf{v}_i + \bar{\varphi}\right)^T \left(\sum_{i=1}^{q} \beta_i \mathbf{v}_i + \bar{\varphi}\right) \\ &= \sum_{i=1}^{q} \beta_i^2 + \bar{\varphi}^T\bar{\varphi} + 2\bar{\varphi}^T \sum_{n=1}^{N} \tilde{\gamma}_n \tilde{\varphi}(\mathbf{x}_n) \\ &= \sum_{i=1}^{q} \left(\sum_{n=1}^{N} \alpha_{in} \tilde{k}(\mathbf{x}, \mathbf{x}_n)\right)^2 + \frac{1}{N^2} \sum_{n,m=1}^{N} k(\mathbf{x}_n, \mathbf{x}_m) \\ &\quad + \frac{2}{N} \sum_{n=1}^{N} \left(\tilde{\gamma}_n \sum_{m=1}^{N} k(\mathbf{x}_m, \mathbf{x}_n) - \frac{\tilde{\gamma}_n}{N} \sum_{m,l=1}^{N} k(\mathbf{x}_m, \mathbf{x}_l)\right) \end{aligned} \tag{9}$$

Finally the last term can be expanded using the same properties as above

$$\begin{aligned} \varphi(\mathbf{z})^T P_q\varphi(\mathbf{x}) &= \varphi(\mathbf{z})^T \left(\sum_{n=1}^{N} \tilde{\gamma}_n(\varphi(\mathbf{x}_n) - \bar{\varphi}) + \bar{\varphi}\right) \\ &= \sum_{n=1}^{N} \gamma_n k(\mathbf{z}, \mathbf{x}_n) \end{aligned} \tag{10}$$

Where the last equality follows from letting $\gamma_n = \tilde{\gamma}_n + \frac{1}{N}(1 - \sum_{j=1}^{N} \tilde{\gamma}_j)$, and where $\tilde{\gamma}_n = \sum_{i=1}^{q} \beta_i \alpha_{in}$ as defined in Eq. 3. Now combining the expressions gives the following cost function

$$\begin{aligned} R(\mathbf{z}) &= ||\varphi(\mathbf{z}) - P_q\varphi(\mathbf{x})||^2 \\ &= k(\mathbf{z}, \mathbf{z}) - 2\sum_{n=1}^{N} \gamma_n k(\mathbf{z}, \mathbf{x}_n) + \Omega \end{aligned} \tag{11}$$

where all the $\mathbf{z}$-independent terms originating from $||P_q\varphi(\mathbf{x})||^2$ have been collected in Ω.

3.1 Overview of Existing Algorithms

The non-linear optimization problem associated with finding the pre-image has been approached in a variety of ways. In the original work by Mika et al. [9, 14] a fixed-point iteration method was proposed. It is a noted drawback of this method that it can be numerically unstable, sensitive to the initial starting point, and converge to a local extremum. To overcome this problem a more 'direct' approach was taken by Kwok and Tsang [8]. They combined the idea of multidimensional scaling (MDS) with the relationship of distance measures in feature space and input space, thereby deriving a non-iterative solution. These are the two approaches most widely used in applications. However,

406 J Sign Process Syst (2011) 65:403–412

several modifications have already been proposed. In order to overcome possible numerical instabilities of the fixed-point approach, various ways of initialization have been suggested. The algorithm can be started in a 'random' input space point, but this can lead to slow convergence in real-life problems, since the cost-function can be very flat in regions away from data. Alternatively, for de-noising applications, it can be initialized in the point in input space, which we seek to de-noise. However, according to Takahashi and Kurita [15] this strategy will only work if the signal-to-noise ratio (SNR) is high. Instead Kim et al. [7] suggested to initialize the fixed-point iteration scheme in the solution found by Kwok and Tsang's direct method. Later it was claimed that a more efficient starting point would be the mean of a certain number of neighbors of the point to be de-noised [16]. Dambreville et al. [4] proposed a modification of the method developed by Mika et al. utilizing feature space distances. This method also minimizes the distance constraint in Eq. 4, but does so in a non-iterative approximation thereby avoiding numerical instabilities. Bakir et al. [2] used kernel ridge regression to learn some inverse mapping of φ. While this formulation is in very general terms, the actual implementation is similar to that of Kwok and Tsang [8]. The main issue is that we typically only have indirect access to feature space points, thus a learned pre-image needs to be formulated in terms of distances as in Kwok and Tsang's method, rather than explicit input-output examples. It should be noted that with the relative general formulation the method of Bakir et al. in some cases can be applied beyond Kwok and Tsang's method, e.g., to non-Euclidean input spaces. In lieu of the recognized ill-posed nature of the inverse problem attempts of more robust estimators have been pursued. Nguyen and De la Torre Frade [10] introduced regularization that penalized the projection in feature space, while Zheng and Lai [20] used a ridge regression regularizer for the weights of a learned pre-image estimator as originally proposed by Bakir et al. [2].

Returning to the iterative scheme of Mika et al., we work, as in most applications, with RBF kernels for which $k(\mathbf{z}, \mathbf{z})$ is constant for all $\mathbf{z}$, hence minimizing the squared distance in Eq. 11 is identical to

$$\max_{\mathbf{z}} 2 \sum_{n=1}^{N} \gamma_n k(\mathbf{z}, \mathbf{x}_n) \tag{12}$$

Now in extrema of Eq. 12 the derivative with respect to $\mathbf{z}$ is zero, which leads to the following fixed-point iteration for a Gaussian kernel of the form $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{c}||\mathbf{x} - \mathbf{x}'||^2\right)$, where c controls the width of the kernel and thereby the non-linearity of the associated feature space map [9]

$$\mathbf{z}_{t+1} = \frac{\sum_{n=1}^{N} \gamma_n \exp(-||\mathbf{z}_t - \mathbf{x}_n||^2/c)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_n \exp(-||\mathbf{z}_t - \mathbf{x}_n||^2/c)} \tag{13}$$

As mentioned maximizing Eq. 12 is a non-linear optimization problem, and hence suffers from convergence to local minima and strong sensitivity to the initial point $\mathbf{z}$. As we shall see, this implies that the solutions are at times highly unstable.

4 Instability Issues

Some of the most recent publications (e.g., [1, 17]) argue that the methods of Mika et al. [9], Kwok and Tsang [8], and Dambreville et al. [4] are the most reliable. In this section we show that these current approaches suffer from different weaknesses.

A distinctive feature of all the algorithms is that they seek to determine the pre-image as a weighted average of the training points. In the method proposed by Kwok and Tsang only k of the training points are used for the estimation, and their weights are based on a distance relation between feature space and input space and the persistence of this distance across the φ-mapping. In Mika et al.'s approach all training points contribute to estimating the pre-image, and the individual weights are found using the pre-image itself, hence the method becomes iterative. Furthermore, the weight of a given training point decays exponentially with input space distance, so only points close to the pre-image contribute significantly to the pre-image estimate. Dambreville et al. substituted the iterative approach by a direct formula, where the weights decrease linearly with feature space distance, giving high weight to training points for which $\varphi(\mathbf{x}_i)$ is close to Ψ.

Thus, in different ways, both Kwok and Tsang's and Dambreville et al.'s method are based on the assumption that points which are close in feature space are also close in input space. For very non-linear kernels this assumption fails. In fact, the more non-linear kernel the more "creased" the associated feature space will be. This is illustrated with a simple 2-dimensional example, where 500 data points are drawn from two rings with Gaussian noise. Kernel PCA is performed using a Gaussian kernel. In order to illustrate how distances are skewed due to the kernel transformation, all pairwise feature space distances are determined, and for the 0.5% closest relations in feature space, the corresponding observation pairs are marked in input space. In Fig. 2a and b these pairs are indicated with

J Sign Process Syst (2011) 65:403–412 407

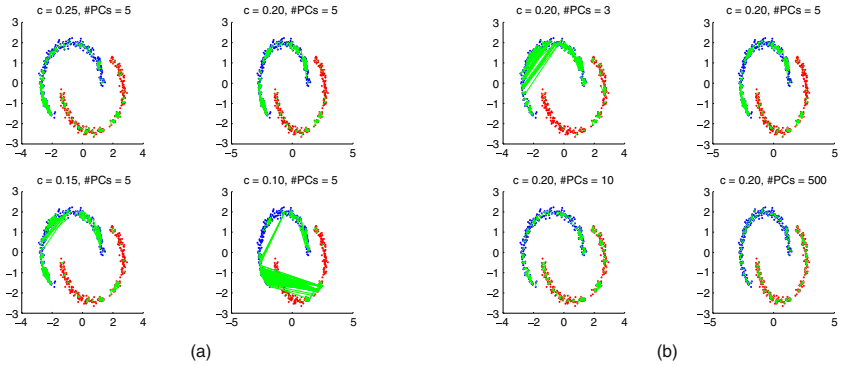


Figure 2 Kernel PCA with a Gaussian kernel is performed on a 2-dimensional two-class example in order to illustrate the distance distortions occurring between input and feature space when either the non-linearity is increased or the principal subspace dimensionality is decreased. The *green lines* indicate the 0.5% closest relations in feature space. The combination of $c = 0.20$ and #PCs $= 5$ is shown in both figures as the *top right plot* for easy comparison. Panel **a** shows how increasing the non-linearity gives rise to unexpected relations in feature space, and hence distortions of the distance relation across the φ-map, while panel **b** illustrates that the same effect occurs when decreasing the dimensionality of the subspace.

green lines. Figure 2b indicates that the distortion is not only affected by the non-linearity of the kernel but also the dimensionality of the principal subspace (i.e., the number of principal components used).

The instability of the fixed-point iteration method is also evident when using a very non-linear kernel. Just like in the previous example, this is illustrated by drawing 500 data points from two partial rings with Gaussian noise. A "noisy" observation, which we seek to de-noise is placed in the center of the rings. Kernel PCA is performed using a Gaussian kernel with varying scale parameter. The number of principal components is fixed to 50. For every scale, Mika et al.'s algorithm is initialized in all training points respectively and the resulting pre-image of the de-noised observation is shown in the top two rows of Fig. 3. Clearly when using a very non-linear kernel, the reconstructed pre-image heavily depends on the initialization. As the cost function has many local minima, the algorithm converges to the nearest one. When c increases the number of distinct pre-images are seen to decrease, until c reaches a certain level, where kernel PCA approaches linear PCA. When this happens the pre-image is drawn towards the noisy observation, as linear PCA fails to capture the non-linear trends clearly visible in the data. In the lower part of Fig. 3 the value of the cost function, $R(z)$ in Eq. 11, are shown for all the found pre-images as a function of the scale. From this figure it is clear that using a very non-linear kernel result in a cost function

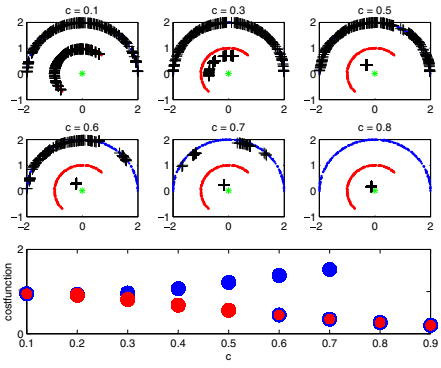


Figure 3 We seek to de-noise the *green point* in $(0, 0)$ using kernel PCA with varying scale and 50 PCs. Mika et. al.'s algorithm is initialized in all 500 training points resulting in 500 pre-image estimates indicated by the *black crosses* in the *two top rows*. The *bottom row* shows the value of the cost function, $R(z)$ in Eq. 11, for each pre-image. The *color* indicates which class the initialization point belongs to. It is clearly seen how a very non-linear kernel results in many local minima with almost the same cost function value, while the more linear case fails to describe the non-linear signal manifold leading to a unique but not denoised pre-image.

408 J Sign Process Syst (2011) 65:403–412

with not only many local minima, but furthermore these minima practically all have the same value. Thereby making iterative algorithms very sensitive to the point of initialization.

Based on the simple examples shown here, it seems reasonable to try to improve the stability of the current approaches. We suggest this is done by introducing regularization in Eq. 11 as further described in the following sections.

5 Regularization

Regularization is commonly used to stabilize estimates of high variability. Thus, if the unregularized criterion is the risk function, $R(\boldsymbol{z})$, the regularized version is obtained by adding a penalty, $T(\boldsymbol{z})$, so that the solution can be formulated as:

$$\boldsymbol{z} = \text{argmin } R(\boldsymbol{z}) + \lambda T(\boldsymbol{z}) \tag{14}$$

where $\lambda > 0$ is the regularization parameter controlling the strength of the penalty term. Hence, Eq. 14 can lead to various estimates depending not only on the chosen penalty term but also on the value of λ.

If both $R(\boldsymbol{z})$ and $T(\boldsymbol{z})$ are differentiable a fixed-point iteration scheme similar to that of Mika et al. can easily be derived.

In this paper we focus on two special cases from the power penalty family, namely Tikhonov regularization [19]

$$\boldsymbol{z} = \text{argmin } R(\boldsymbol{z}) + \lambda ||\mathbf{z} - \mathbf{x}_0||^2_{\ell_2} \tag{15}$$

and the Lasso [18] where

$$\boldsymbol{z} = \text{argmin } R(\boldsymbol{z}) + \lambda ||\mathbf{z}||_{\ell_1} \tag{16}$$

While Tikhonov regularization stabilizes the estimate, Lasso regularization will for sufficiently large values of λ force some of the z_j's to zero, leading to a sparse pre-image. Notice also that the Lasso problem can be interpreted as a MAP estimate with a Laplacian prior on the $\boldsymbol{z}_i$'s.

The type of penalty should be chosen according to the given problem, and hence, prior knowledge of the expected pre-image will often work as the base for choosing the penalty term.

5.1 Tikhonov Input Space Regularization

In order to provide a more stable estimate of the pre-image we propose to augment the cost function with an input space distance penalty term (see Fig. 1)

$$\begin{aligned}\rho_1(\mathbf{z}) &= R(\boldsymbol{z}) + \lambda T_2(\boldsymbol{z}) \\ &= ||\varphi(\mathbf{z}) - P_q\varphi(\mathbf{x})||^2 + \lambda||\mathbf{z} - \mathbf{x}_0||^2_{\ell_2} \\ &= k(\mathbf{z},\mathbf{z}) - 2\sum_{n=1}^{N}\gamma_n k(\mathbf{z},\mathbf{x}_n) + \Omega \\ &\quad +\lambda(\mathbf{z}^T\mathbf{z} + \mathbf{x}_0^T\mathbf{x}_0 - 2\mathbf{z}\mathbf{x}_0)\end{aligned} \tag{17}$$

where $\mathbf{x}_0$ is the noisy observation in $\mathcal{X}$. The main rationale is that among the solutions to the non-linear optimization problem we want the pre-image which is closest to the noisy input point, hence, hopefully reduce possible distortions of the signal. Thus we seek to minimize

$$\rho_2(\mathbf{z}) = k(\mathbf{z},\mathbf{z}) - 2\sum_{n=1}^{N}\gamma_n k(\mathbf{z},\mathbf{x}_n) + \lambda(\mathbf{z}^T\mathbf{z} - 2\mathbf{z}\mathbf{x}_0) \tag{18}$$

ignoring all $\mathbf{z}$-independent terms. This expression can be minimized for any kernel using a non-linear optimizer.

For RBF kernels the fixed-point iteration scheme can be regularized similarly, this typically leads to a faster evaluation than using an optimizer. Introducing regularization in the maximization problem given in Eq. 12 leads to the following objective function

$$\rho_3(\mathbf{z}) = 2\sum_{n=1}^{N}\gamma_n k(\mathbf{z},\mathbf{x}_n) - \lambda||\mathbf{z} - \mathbf{x}_0||^2 \tag{19}$$

which we seek to maximize with respect to $\mathbf{z}$. With straightforward algebra we get the regularized fixed-point iteration

$$\mathbf{z}_{t+1} = \frac{\frac{2}{c}\sum_{n=1}^{N}\gamma_n \exp\left(-\frac{1}{c}||\mathbf{z}_t - \mathbf{x}_n||^2\right)\mathbf{x}_n + \lambda\mathbf{x}_0}{\frac{2}{c}\sum_{n=1}^{N}\gamma_n \exp\left(-\frac{1}{c}||\mathbf{z}_t - \mathbf{x}_n||^2\right) + \lambda} \tag{20}$$

In this expression the denominator is given by $\frac{2}{c}\langle\varphi(\mathbf{z}_t), \Psi\rangle + \lambda$. As λ is a non-negative parameter, the denominator will always be non-zero in the neighborhood of a maximum because the inner-product will be positive in that same neighborhood.

5.2 Sparse Reconstruction by Lasso Regularization

In many applications, introducing other types of regularization seems appealing. We will in the following

J Sign Process Syst (2011) 65:403–412 409

show how ℓ_1-norm regularization improves the performance, when the sought pre-image is expected to be sparse, i.e., only a fraction of the input dimensions are nonzero. Hence, we now seek

$$\mathbf{z} = \underset{\mathbf{z}\in\mathcal{X}}{\operatorname{argmin}} \; ||\varphi(\mathbf{z}) - P_q\varphi(\mathbf{x}_0)||^2 + \lambda||\mathbf{z}||_{\ell_1} \tag{21}$$

Which can be reformulated as minimizing the following cost function

$$\begin{aligned}\rho_4(\mathbf{z}) &= R(\mathbf{z}) + \lambda T_1(\mathbf{z}) \\ &= ||\varphi(\mathbf{z}) - P_q\varphi(\mathbf{x})||^2 + \lambda||\mathbf{z}||_{\ell_1} \\ &= -2\sum_{n=1}^{N}\gamma_n k(\mathbf{z},\mathbf{x}_n) + \lambda\sum_{j=1}^{D}|z_j|\end{aligned} \tag{22}$$

where the last equality only holds for RBF kernels.

Since $T_1(\boldsymbol{z})$ is not differentiable, implementing a fixed-point iteration scheme is not feasible. Instead we will apply the generalized path seeking (GPS) framework as introduced by Friedman in [5] to estimate both the pre-image and the degree of regularization simultaneously.

In order to calculate the regularization parameter path, λ, and drive the GPS algorithm, the following first order derivatives are needed:

$$\frac{dR}{dz_j} = 2\sum_{n=1}^{N}\gamma_n \exp\left(-\frac{1}{c}||\mathbf{z}-\mathbf{x}_n||^2\right)\cdot\frac{2(z_j - x_{jn})}{c} \tag{23}$$

and

$$\frac{dT_1}{d|z_j|} = 1 \tag{24}$$

Applying the GPS algorithm is now straight forward. Friedman suggest using an adaptive step length when exploring the solution space, however, for simplicity we chose a fixed step length of $1e^{-2}\max(|\boldsymbol{X}|)$ in the following experiments. This is a very conservative step length, and may be tuned for faster convergence. The algorithm is stopped when the cost function stabilizes.

It is noted, that the GPS algorithm could also be used to get an indication of the magnitude of the regularization needed in the fixed point iteration scheme. However, as the GPS framework needs many more iterations than the fixed-point scheme, this way of estimation is not attractive for general estimation.

For further background on smoothing by ℓ_p norms we refer the reader to [11].

6 Experiments

In this section we compare the new regularization approaches to the existing methods proposed by: (a) Kwok-Tsang [8], (b) Dambreville et al. [4], and (c) Mika et al. [9]. The experiments are done on a subset of the USPS data consisting of 16×16 pixels handwritten digits.[2] For each of the digits 0, 2, 4, and 9 we chose 100 examples for training and another 100 examples for testing. We added Gaussian noise $\mathcal{N}(0, 0.25)$ and set the regularization parameter in Eq. 19 to $\lambda = 3e^{-4}$.

In order to illustrate the stability and performance of the methods, we vary both the number of principal components used to define the signal manifold and the scale parameter c of the Gaussian kernel. For each combination and pre-image estimator, the mean squared error (MSE) of the de-noised result for the 400 *test* examples is calculated. The iterative approaches are initialized in the noisy test point and for the Kwok and Tsang's approach 10 neighbors were used for the approximation.

The results are summarized in Fig. 4 where we show the lower 5th and upper 95th percentile confidence intervals for the MSE. In order to ease the comparison and adjust for potential bias in the estimation, all pre-images are re-normalized to the range of the original image before the squared error is calculated. As seen the confidence intervals blow up for the previous methods—panels (a–c)—in the non-linear regime in which the kernel has a relative small scale parameter. At the same time the confidence interval points to a much more stable de-noised solution for the new Tikhonov input space regularized approach—as seen in panel (d).

To better understand the nature of the instability of the previous algorithms we have investigated the diversity of the solutions obtained when starting the fixed-point iterative algorithms in different initial points. Specifically we compare the standard iterative solution of Mika et al. and the new input space regularized version. For each of the 400 test examples the two algorithms are initialized in 40 randomly chosen training examples. This leads to 40 (potentially different) pre-image solutions for each test sample. We measure the stability of these solution sets as the mean pairwise distance between them 40 pre-images, and report the mean across the 400 test examples This mean and its confidence intervals are presented in Fig. 5 as a function of the non-linearity scale parameter c. As

[2]The USPS data set is described in [6] and can be downloaded from www.kernel-machines.org.

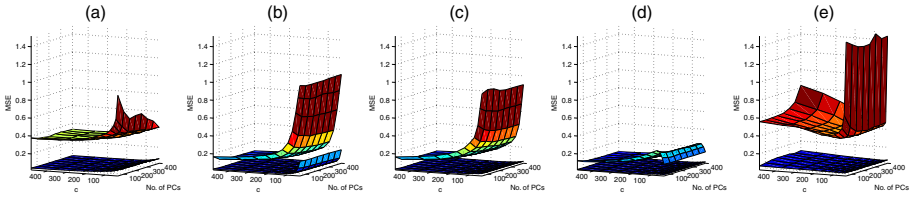


Figure 4 Experiment to illustrate the stability of pre-image based de-noising of USPS digits. A training set of 400 digits (100@0, 2, 4, 9) is used to define the signal manifold. We show the confidence intervals (5th and the 95th percentile) for the mean square error (MSE) in different combinations of kPCA subspace dimension and non-linearity. MSE computed for 400 de-noised test samples for (**a**) Kwok-Tsang, (**b**) Dambreville et al., (**c**) Mika et al., (**d**) using Tikhonov input space regularization, and (**e**) using Lasso regularization. The previous schemes are seen to deteriorate in the non-linear regime (*small c*) compared to the input space regularization approach.

seen, the Tikhonov input space regularization approach produces a stable pre-image even for very non-linear models (small c), where the un-regularized iterative scheme fails to reproduce.

Finally Fig. 6 shows visual examples of the de-noised images obtained with Mika et al.'s and the two new regularized approaches. For the images which are successfully de-noised by Mika et al.'s method, e.g., some of the 'zeros' or 'nines', adding regularization has very little effect. However, a clear improvement can be seen for the images for which Mika et al.'s algorithm fails to recover a good visual solution. For these digits the input space regularization method do reconstruct the correct digit, albeit with a price paid in terms of a slightly less de-noised result. Furthermore, the image intensity, as shown in the lower part of Fig. 6, illustrates the increased SNR achieved by the input space regularization.

The last panel in Fig. 6 illustrates the effect of regularizing pre-image estimation by the sparsity promoting Lasso penalty. The majority of the digits are clearly identifiable and only a minimum of background noise

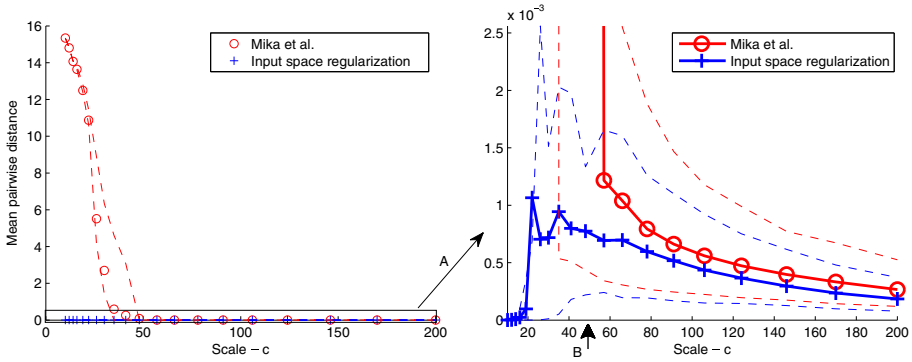


Figure 5 Illustration of the instability. The mean pairwise distances between solutions obtained after initializing in 40 randomly chosen training set input points (mean, 5th and the 95th percentiles) for Mika et al. (*red*) and the new Tikhonov input space regularization approach (*blue*). We use 300 principal components in this study. The previous approach fails to provide a stable pre-image in the non-linear regime (*small c*). The *right panel* is a close-up of the box indicated on the *left panel*. Arrow '**B**' indicates the scale used in Fig. 6.

J Sign Process Syst (2011) 65:403–412 411

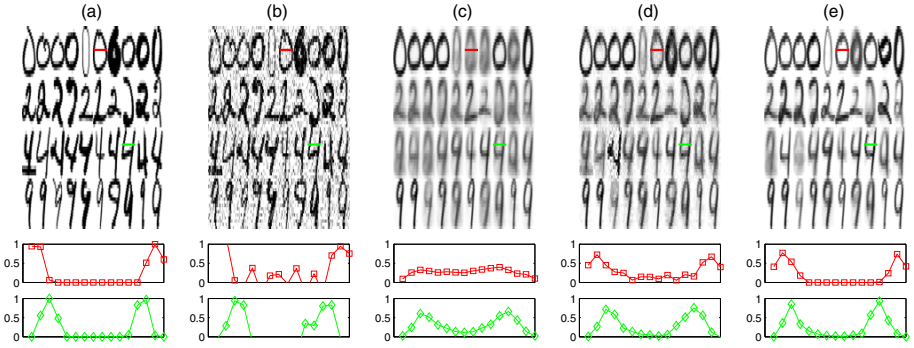


Figure 6 *Top*: Example of de-noised digits using a very non-linear kernel ($c = 50$) and 100 principal components. The colormap has been adjusted for better visualization. *Bottom*: The image intensity along the 16 pixel segments indicated by the *red* and the *green line* in the *upper panels*. Panel **a** shows the original digits, **b** shows the digits after Gaussian noise have been added, **c** is the de-noised digits by Mika et al.'s algorithm, **d** de-noising using Tikhonov regularization, and **e** using sparse reconstruction by Lasso regularization. Note the improved SNR in the results of the new methods.

is present. Again the high SNR is reflected in the lower part of the figure. However, as noted in Fig. 4 sparse reconstruction does not lead to a stable estimation in terms of the MSE measure.

7 Conclusion

In this contribution we addressed the problem of pre-image instability for kernel PCA de-noising. The recognized concerns of current methods, e.g., the sensitivity to local minima and large variability were demonstrated for the most widely used methods including Mika et al.'s iterative scheme, Kwok-Tsang's local linear approximation and the method of Dambreville et al. By introducing simple input space distance regularization in the existing pre-image approximation cost function, we achieved a more stable pre-image, with very little sacrifice of the de-noising ability. Experimental results on the USPS data illustrated how input space regularization provides a more stable pre-image in the sense of variability between test points and reduced the sensitivity to starting conditions as well as provided a better visual result. Furthermore, we introduced ℓ_1-norm Lasso regularization and demonstrated that it leads to an improved estimate in terms of visual quality. This regularizer however incurs a relatively large mean squared error in the data set investigated here. The trade-off between quantitative and qualitative performances of the two regularizers needs further investigation.

We thus recommend to augment the cost function for pre-image estimation in Eq. 11 with a task specific penalty term. When the aim is superior visual quality, and the data is known to be sparse, sparse reconstruction by Lasso regularization should be employed. In cases where the objective is both visual impression and stability of the estimate, we suggest the use of Tikhonov input space distance regularization as it provides a reliable pre-image in cases where current methods fail to recover a meaningful result.

In future work we aim to combine input space regularization with sparse reconstruction in order to achieve both highly stabile and attractive visual results as well as extend the concept of sparse reconstruction in relation to kernel methods. Furthermore, the amount of regularization is to be investigated further for both methods presented in this paper.

Acknowledgement This work is supported in part by the Lundbeckfonden through the Center for Integrated Molecular Brain Imaging (Cimbi), www.cimbi.org.

References

1. Arias, P., Randall, G., & Sapiro, G. (2007). Connecting the out-of-sample and pre-image problems in kernel methods. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 1–8). Minneapolis, MN.
2. Bakir, G. H., Weston, J., & Schölkopf, B. (2004). Learning to find pre-images. In S. L. Thrun, & B. Schölkopf (Eds.), *Advances in neural information processing systems* (Vol. 16, pp. 449–456). Cambridge: MIT.

3. Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.
4. Dambreville, S., Rathi, Y., & Tannenbaum, A. (2006). Statistical shape analysis using kernel PCA. In *IS&T/SPIE symposium on electrical imaging*.
5. Friedman, J. H. (2008). *Fast sparse regression and classification*. Tech. rep., Department of Statistics, Stanford University.
6. Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(5), 550–554.
7. Kim, K. I., Franz, M. O., & Schölkopf, B. (2005). Iterative kernel principal component analysis for image modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(9), 1351–1366.
8. Kwok, J. T. Y., & Tsang, I. W. H. (2004). The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks, 15*(6), 1517–1525.
9. Mika, S., Schölkopf, B., Smola, A., Müller, K. R., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems* (Vol. 11, pp. 536–542). Cambridge: MIT.
10. Nguyen, M. H., & la Torre, F. D. (2009). Robust kernel principal component analysis. *Advances in Neural Information Processing Systems, 21*, 1185–1192.
11. Rice, J. R., & White, J. S. (1964). Norms for smoothing and estimation. *SIAM Review, 6*, 243–256.
12. Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K. R., Rätsch, G., et al. (1999). Input space versus feature space in kernel-based methods. *IEEE Transactions On Neural Networks, 10*(5), 1000–1017.
13. Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*(5), 1299–1319.
14. Schölkopf, B., Smola, A. J., Knirsch, P., & Burges, C. J. C. (1998). Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. In P. Levi, M. Schanz, R.-J. Ahlers, & F. May (Eds.), *Mustererkennung 1998, 20. DAGM-Symposium* (pp. 125–132). Berlin: Springer.
15. Takahashi, T., & Kurita, T. (2002). Robust de-noising by kernel PCA. In *ICANN '02: Proceedings of the international conference on artificial neural networks* (pp. 739–744). Berlin: Springer.
16. Teixeira, A. R., Tomé, A. M., Stadlthanner, K., & Lang, E. W. (2008). KPCA denoising and the pre-image problem revisited. *Digital Signal Processing, 18*(4), 568–580.
17. Thorstensen, N., Segonne, F., & Keriven, R. (2008). Normalization and preimage problem in gaussian kernel PCA. In *2008 15th IEEE international conference on image processing—ICIP 2008*, (pp. 741–744). San Diego, California.
18. Tibshirani, R. (1994). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B, 58*, 267–288.
19. Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solution of ill-posed problems*. New York: Wiley.
20. Zheng, W. S., & Lai, J. H. (2006). Regularized locality preserving learning of pre-image problem in kernel principal component analysis. *International Conference on Pattern Recognition, 2*, 456–459.

Trine Julie Abrahamsen received the B.Sc. degree in Biomedical Engineering in 2007 and the M.Sc. degree in Mathematical Modeling and Computation in 2009 both from the Technical University of Denmark, Denmark. She is now pursuing a PhD degree in Applied Mathematics at the Department of Informatics and Mathematical Modeling, Technical University of Denmark. Her research interests include machine learning for life science applications, with special focus on kernel methods.

Lars Kai Hansen is a professor of digital signal processing at the Technical University of Denmark, Lyngby, were he also heads the Cognitive Systems Section. His research concerns adaptive signal processing and machine learning with applications in biomedicine and digital media. He has published more than 225 contributions on these subjects in journals, conferences, and books.

Appendix B

Sparse non-linear denoising: Generalization performance and pattern reproducibility in functional MRI

Trine Julie Abrahamsen and Lars Kai Hansen. Sparse Non-linear Denoising: Generalization Performance and Pattern Reproducibility in Functional MRI. *Pattern Recognition Letters*, 32(15):2080 – 2085, 2011.
doi:10.1016/j.patrec.2011.08.006.

Pattern Recognition Letters 32 (2011) 2080–2085

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Sparse non-linear denoising: Generalization performance and pattern reproducibility in functional MRI

Trine Julie Abrahamsen *, Lars Kai Hansen

DTU Informatics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 28 April 2011
Available online 23 August 2011
Communicated by M.A. Girolami

Keywords:
Kernel PCA
Pre-image estimation
Denoising
Sparsity
Reproducibility
Functional MRI

ABSTRACT

We investigate sparse non-linear denoising of functional brain images by kernel principal component analysis (kernel PCA). The main challenge is the mapping of denoised feature space points back into input space, also referred to as "the pre-image problem". Since the feature space mapping is typically not bijective, pre-image estimation is inherently illposed. In many applications, including functional magnetic resonance imaging (fMRI) data which is the application used for illustration in the present work, it is of interest to denoise a sparse signal. To meet this objective we investigate sparse pre-image reconstruction by Lasso regularization. We find that sparse estimation provides better brain state decoding accuracy and a more reproducible pre-image. These two important metrics are combined in an evaluation framework which allow us to optimize both the degree of sparsity and the non-linearity of the kernel embedding. The latter result provides evidence of signal manifold non-linearity in the specific fMRI case study.

1. Introduction

Multivariate denoising by projection onto linear subspaces such as in principal component analysis (PCA) or independent component analysis takes advantage of linear spatio-temporal dependencies in the wanted data. However, more general dependency structures can be exploited is we invoke denoising based on projection onto non-linear manifolds. In the present work we will use functional magnetic resonance imaging (fMRI) as a case study, this domain is not only of significant importance to neuroscience but has also become and important benchmark for pattern recognition and machine learning (Mørch et al., 1997; Pereira et al., 2009). Non-linear modeling of fMRI by *kernel* PCA was first considered by Thirion and Faugeras (2003) for separation of dynamical components. In (Hansen, 2007) we presented evidence that non-linear embedding of fMRI may lead to improved decoding performance over denoising by conventional linear PCA, thus the decoding relevant manifolds could be non-linear.

The practical use of non-linear kernel based pre-processing methods has been limited. Wider acceptance has likely been obstructed by a number of methodological issues including lack of stable visualization schemes, see e.g., Abrahamsen and Hansen (2010), and the lack of tools for tuning of control parameters. These issues have largely been solved in linear denoising using data resampling schemes that allow estimation of both out-of-sample generalizability (Hansen et al., 1999) and pattern reproducibility (Yourganov et al., 2011). In this present work we aim to generalize these methods to non-linear denoising.

The basic idea of denoising by kernel PCA is to implement a projection onto a signal manifold in three steps, in the first step we map the original input space data into a feature space in which the manifold is linearized. The second step then consists of using a conventional linear algorithm, like PCA, to identify the signal manifold by linear projection in feature space. Finally, in the third step we estimate the *denoised* input space point that best represent the projected feature space point. The latter step is referred to as the *pre-image problem*. Reconstructing a reliable pre-image is challenging and has given rise to several algorithms (Bakir et al., 2004; Dambreville et al., 2006; Kwok and Tsang, 2004; Mika et al., 1999). We have previously shown that regularized pre-image estimation is more stable and also presented *initial evidence* that sparse reconstruction may improve visual quality of denoised patterns (Abrahamsen and Hansen, 2010).

In this work we show how sparse pre-image reconstruction improves performance relative to dense reconstruction in terms of both brain state predictability and brain map reproducibility in the context of fMRI analysis. Furthermore, optimizing for both predictability and reproducibility provides a means for tuning of sparse kernel PCA control parameters.

2. Kernel PCA and robust pre-image estimation

To set the stage for our discussion of sparse pre-images, let us first review the most salient aspects of kernel PCA denoising (Mika et al., 1999; Schölkopf et al., 1998). Let $\mathcal{F}$ be the Reproducing

* Corresponding author.
E-mail address: tjab@imm.dtu.dk (T.J. Abrahamsen).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.08.006

Kernel Hilbert Space (RKHS) associated with the kernel function $k(\boldsymbol{x},\boldsymbol{x}') = \varphi(\boldsymbol{x})^T\varphi(\boldsymbol{x}')$, where $\varphi : \mathcal{X} \mapsto \mathcal{F}$ is a non-linear map from the D-dimensional input space $\mathcal{X}$ to the possibly infinite dimensional feature space $\mathcal{F}$. Here and in the following we use the notation: bold uppercase letters denote matrices, bold lowercase letters represent column vectors, and non-bold letters denote scalars. $\boldsymbol{a}_j$ denotes the jth column of $\boldsymbol{A}$, while a_{ij} is the matrix element in the ith row and jth column of $\boldsymbol{A}$. Finally, $\boldsymbol{1}_{NN'}$ is an $N \times N'$ matrix of ones. Let $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ be N training data points in $\mathcal{X}$ and $\{\varphi(\boldsymbol{x}_1),\ldots,\varphi(\boldsymbol{x}_N)\}$ be the corresponding feature images in $\mathcal{F}$. The mean of the φ-mapped data points is denoted $\bar{\varphi} = \frac{1}{N}\sum_{n=1}^{N}\varphi(\boldsymbol{x}_n)$ and the mean centered features are given by $\tilde{\varphi}(\boldsymbol{x}) = \varphi(\boldsymbol{x}) - \bar{\varphi}$. Let $\boldsymbol{K}$ denote the kernel matrix with element $K_{ij} = k(\boldsymbol{x}_i,\boldsymbol{x}_j)$, then kernel PCA can be performed by solving the eigenvalue problem

$$\widetilde{\boldsymbol{K}}\boldsymbol{\alpha}_i = \lambda_i\boldsymbol{\alpha}_i, \tag{1}$$

where $\widetilde{\boldsymbol{K}}$ is the mean centered kernel matrix given by

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \frac{1}{N}\boldsymbol{1}_{NN}\boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\boldsymbol{1}_{NN} + \frac{1}{N^2}\boldsymbol{1}_{NN}\boldsymbol{K}\boldsymbol{1}_{NN}. \tag{2}$$

The projection of a feature mapped test point onto the ith principal component is

$$\beta_i(\boldsymbol{x}) = \tilde{\varphi}(\boldsymbol{x})^T\boldsymbol{v}_i = \sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\boldsymbol{x})^T\tilde{\varphi}(\boldsymbol{x}_n) = \sum_{n=1}^{N}\alpha_{in}\tilde{k}(\boldsymbol{x},\boldsymbol{x}_n), \tag{3}$$

where $\boldsymbol{v}_i = \sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\boldsymbol{x}_n)$ is the ith eigenvector of the feature space covariance matrix and the $\boldsymbol{\alpha}_i$'s are normalized to unit length. For denoising applications we are interested in projection onto the signal subspace, in particular the feature subspace spanned by the leading eigenvectors of the covariance matrix. The projection of $\varphi(\boldsymbol{x})$ onto the subspace spanned by the first q eigenvectors will be denoted $P_q\varphi(\boldsymbol{x})$ and are given by

$$\begin{aligned} P_q\varphi(\boldsymbol{x}) &= \sum_{i=1}^{q}\beta_i\boldsymbol{v}_i + \bar{\varphi} = \sum_{i=1}^{q}\beta_i\sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\boldsymbol{x}_n) + \bar{\varphi} \\ &= \sum_{n=1}^{N}\tilde{\gamma}_n\tilde{\varphi}(\boldsymbol{x}_n) + \bar{\varphi}, \end{aligned} \tag{4}$$

where $\tilde{\gamma}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$. Kernel PCA satisfies properties similar to those for linear PCA, namely that the squared reconstruction error is minimal and the retained variance is maximal. However, these properties hold in $\mathcal{F}$ not in $\mathcal{X}$. The rationale behind kernel PCA is that the kernel mapping linearize the relevant manifolds, hence, makes a non-linear problem amenable to conventional linear denosing techniques, for a more detailed discussion of this important point, see Roweis and Saul (2000).

For denoising purposes it is of interest to reconstruct a data point in input space that corresponds to a specific linearly denoised point in feature space, hence, applying the inverse map of φ. Thus, given such a point, Ψ, we are interested in finding a point $\boldsymbol{z} \in \mathcal{X}$ such that $\varphi(\boldsymbol{z}) = \Psi$ and we will call $\boldsymbol{z}$ the pre-image of Ψ (Schölkopf et al., 1998). For many non-linear kernels $\dim(\mathcal{F}) \gg \dim(\mathcal{X})$ and φ is neither surjective nor injective. As a function $f : X \mapsto Y$ has an inverse if and only if it is bijective, we do not expect φ to have a well defined inverse. The fact that φ is not surjective implies that there can be points in $\mathcal{F}$ and also in the span of $\{\varphi(\mathcal{X})\}$ that are not images of any $\boldsymbol{x} \in \mathcal{X}$. Finally, when φ is not injective, recovered pre-images are not expected to be unique. In conclusion the pre-image problem is an ill-posed inverse problem (Arias et al., 2007; Burges, 1998; Dambreville et al., 2006; Kwok and Tsang, 2004) and we follow Mika et al. (1999) and relax the problem to that of finding an *approximate* pre-image, i.e., a point in input space which maps into a point in feature space 'as close as possible' to Ψ (see Fig. 1). To implement this search we seek to minimize the distance between $\varphi(\boldsymbol{z})$ and Ψ with respect to $\boldsymbol{z}$. Further assuming that Ψ lies in (or close to) the span of $\{\varphi(\boldsymbol{x}_i)\}$, Ψ can be represented as a linear combination of the training images, i.e. $P_q\varphi(\boldsymbol{x})$, without loss of generality. Thus, we use a quadratic objective function, which can be simplified as

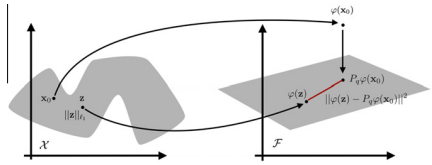


Fig. 1. The pre-image problem in kernel PCA denoising concerns estimating $\boldsymbol{z}$ from $\boldsymbol{x}_0$, through the projection of the image onto the principal subspace in feature space, $\mathcal{F}$.

$$R(\boldsymbol{z}) = \|\varphi(\boldsymbol{z}) - P_q\varphi(\boldsymbol{x})\|^2 = k(\boldsymbol{z},\boldsymbol{z}) - 2\sum_{n=1}^{N}\gamma_n k(\boldsymbol{z},\boldsymbol{x}_n) + \Omega, \tag{5}$$

where $\boldsymbol{z}$-independent terms are collected in Ω, and we have defined $\gamma_n = \tilde{\gamma}_n + \frac{1}{N}\left(1 - \sum_{j=1}^{N}\tilde{\gamma}_j\right)$.

In line with most earlier work, we implement kernel PCA in terms of radial basis function (RBF) kernels for which $k(\boldsymbol{z},\boldsymbol{z})$ is a constant, hence, the objective further simplifies to

$$R(\boldsymbol{z}) = -2\sum_{n=1}^{N}\gamma_n k(\boldsymbol{z},\boldsymbol{x}_n) + \Omega'. \tag{6}$$

The minima of (6) are among points in which the derivative with respect to $\boldsymbol{z}$ is zero, which leads to the following fixed-point iteration for a Gaussian RBF kernel of the form $k(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{1}{c}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$, where c controls the width of the kernel and thereby the non-linearity of the associated feature space map (Mika et al., 1999)

$$\boldsymbol{z}_{t+1} = \frac{\sum_{n=1}^{N}\gamma_n\exp(-\|\boldsymbol{z}_t - \boldsymbol{x}_n\|^2/c)\boldsymbol{x}_n}{\sum_{n=1}^{N}\gamma_n\exp(-\|\boldsymbol{z}_t - \boldsymbol{x}_n\|^2/c)}. \tag{7}$$

The cost in (6) may be highly multi modal, leading to a non-linear optimization problem, and hence the fix point iteration scheme can suffer from convergence to local minima. This typically implies sensitivity to the initial point $\boldsymbol{z}$ and leads to significant instability of the denoising solution, see e.g., Abrahamsen and Hansen (2010) for further illustration of the relations between the feature map non-linearity and pre-image variability.

2.1. Sparse reconstruction by Lasso regularization

High variance can often be suppressed by proper regularization. Here we explore this solution for pre-image stabilization. Let the un-regularized criterion be the risk function, $R(\boldsymbol{z})$. Adding the penalty term, $T(\boldsymbol{z})$, the regularized optimization problem can be formulated as

$$\boldsymbol{z} = \operatorname{argmin} R(\boldsymbol{z}) + \lambda T(\boldsymbol{z}), \tag{8}$$

where $\lambda > 0$ is a regularization control parameter. For differentiable $R(\boldsymbol{z})$ and $T(\boldsymbol{z})$ we derived a fixed-point iteration scheme in (Abrahamsen and Hansen, 2010) and obtained a simple generalization of Eq. (7) leading to more robust pre-image estimates.

In this paper we focus on a special case from the power penalty family, namely the Lasso (Tibshirani, 1994) where (as illustrated in Fig. 1)

$$\boldsymbol{z} = \operatorname{argmin} R(\boldsymbol{z}) + \lambda\|\boldsymbol{z}\|_{\ell_1}. \tag{9}$$

Lasso regularization will for sufficiently large values of λ force all of the z_j's to zero, while for smaller λ a sparse pre-image is obtained in which some z_j's are non-zero. For additional background on smoothing by ℓ_p norms we refer the reader to Rice and White (1964). We notice that the Lasso regularizer and the minimization problem in (9) may be interpreted as a *Maximum A Posteori* estimate with individual Laplacian priors on the pre-image features z_j. With the Lasso regularizer we are interested in the problem

$$\begin{aligned} \boldsymbol{z} &= \underset{\boldsymbol{z}\in\mathcal{X}}{\operatorname{argmin}} R(\boldsymbol{z}) + \lambda T_1(\boldsymbol{z}) \\ &= \underset{\boldsymbol{z}\in\mathcal{X}}{\operatorname{argmin}} \|\varphi(\boldsymbol{z}) - P_q\varphi(\boldsymbol{x}_0)\|^2 + \lambda\|\boldsymbol{z}\|_{\ell_1}. \end{aligned} \tag{10}$$

Which for RBF kernels can be formulated as minimizing the following cost function

$$\rho(\boldsymbol{z}) = -2\sum_{n=1}^{N}\gamma_n k(\boldsymbol{z},\boldsymbol{x}_n) + \lambda\sum_{j=1}^{D}|z_j| + \Omega'. \tag{11}$$

However, since $T_1(\boldsymbol{z})$ is not differentiable we cannot simply get a fixed-point iteration scheme as for quadratic regularization.

As suggested and briefly explored in (Abrahamsen and Hansen, 2010) we can apply the generalized path seeking (GPS) framework introduced by Friedman (2008) to estimate a sparse pre-image. Pseudo-code for applying the GPS algorithm for sparse pre-image reconstruction is given in Algorithm 1. The GPS framework starts with strong regularization, and reduces it through a series of estimates of decreasing sparsity, providing solutions with multiple degrees of sparsity along the path. In order to calculate the regularization parameter path, λ, and drive the GPS algorithm, the following first order derivative is needed:

$$\frac{dR}{dz_j} = 2\sum_{n=1}^{N}\gamma_n \exp\left(-\frac{1}{c}\|\boldsymbol{z}-\boldsymbol{x}_n\|^2\right)\frac{2(z_j - x_{jn})}{c}. \tag{12}$$

Friedman suggests a scheme with an adaptive step length, while we for simplicity have chosen a fixed step length of $\Delta v = 5e^{-2}\|\boldsymbol{X}\|_\infty$ in the our experiments. This is a conservative choice of step length, and may be tuned for faster convergence. The algorithm is stopped when the pre-image estimate stabilizes.

Algorithm 1: GPS for pre-image estimation

1: Initialize: $i = 1$ and $\boldsymbol{z}^1 = \boldsymbol{0}$
2: **repeat**
3: Compute $\left\{\lambda_j^i = -\frac{dR}{dz_j}\right\}_{j=1}^{N}$ using Eq. (12)
4: $S = \left\{j | \lambda_j^i z_j^i < 0\right\}$ {non-zero elements with sign constraint}
5: Find j^* {direction to move in}
6: **if** $S = \emptyset$ **then**
7: $j^* = \operatorname{argmax}_{j=\{1,2,\ldots,N\}}\left|\lambda_j^i\right|$
8: **else**
9: $j^* = \operatorname{argmax}_{j\in S}\left|\lambda_j^i\right|$
10: **end if**
11: Let $\boldsymbol{z}^{i+1} = \boldsymbol{z}^i$
12: $z_{j^*}^{i+1} = z_{j^*}^i + \operatorname{sign}\left(\lambda_{j^*}^i\right)\Delta v$ {update $\boldsymbol{z}$ estimate in j^*th direction}
13: $i = i + 1$
14: **until** $\|\boldsymbol{\lambda}^i\|_\infty < \epsilon$ or $\boldsymbol{z}$ estimate stabilizes

At line 1 the algorithm is initialized in the fully sparse solution. For each step the λ vector is then computed at line 3. Since, $\frac{dT_1}{dz_j} = 1$ the estimation is simpler than shown by Friedman (2008). In order to determine which dimension (j^*) of $\boldsymbol{z}^i$ to update, line 4 identifies the set, S, of non-zero coefficients of $\boldsymbol{z}^i$ with opposite sign to the corresponding element of λ^i. If S is empty, the dimension corresponding to the largest absolute value of λ^i is selected. If S is not empty, j^* is chosen similarly, however, only considering indices from the subset, S. This is shown in line 6 thru 10. At line 12 the selected dimension is increased with the step length, Δv, in the direction of the sign of $\lambda_{j^*}^i$, while all other dimensions remain unchanged (line 11).

Since the set S will be empty when $\boldsymbol{z}^i = \boldsymbol{0}$ or if for all $j : z_j^i \neq 0, \lambda_j^i z_j^i > 0$, the update, $z_{j^*}^{i+1} = z_{j^*}^i + \operatorname{sign}\left(\lambda_{j^*}^i\right)\Delta v$ will always lead to an increase in the absolute value of z_{j^*} in this case. On the other hand, the update will result in a decrease in the absolute value of z_{j^*} only when S is non-empty, since j^* is chosen among directions for which $\lambda_j^i z_j^i < 0$ in this case. Since, S will be empty in most cases (Friedman, 2008), consecutive decreases in the absolute value of z_k is unlikely. Hence, once a dimension of $\boldsymbol{z}$ has become non-zero it will rarely reduce to zero in further iterations, and thus, the estimate of $\boldsymbol{z}$ will move from a fully sparse reconstruction to a fully dence estimate of $\boldsymbol{z}$ unless the algorithm is terminated prior to that due to a small derivative of $R(\boldsymbol{z})$ or when the $\boldsymbol{z}$ estimate has stabilized.

2.2. Measuring the predictive value and robustness of denoising

In many application domains including functional neuroimaging, statistical learning has two equally important objectives: (1) An accurate predictive model; (2) A robust interpretation of the underlying physical mechanism that allows prediction. In the context of functional neuroimaging by fMRI the first objective can be evaluated, e.g., in terms of the brain state decoding accuracy (Mørch et al., 1997), while the latter objective can be quantified by the reproducibility of visualization, the brain map (Strother et al., 2002) (NPAIRS). Here we follow Strother et al. (2002) and use split-half resampling to produce unbiased estimates of the variability of denoised pre-images. The reproducibility of the reconstructed pre-image is computed by denoising a given datum based on each of the two half samples. As the split is random the two halves are statistical exchangeable. Hence, the squared difference is an unbiased estimate of the sampling variance of the whole non-linear procedure. The overall reproducibility is represented by the correlation coefficient between the two reconstructed pre-images. By repeating the split-half procedure we can obtained stable unbiased estimates of reproducibility. Brain state decoding performance is evaluated in cross-validation using a simple linear discriminant in the q-dimensional signal feature space. Following the NPAIRS procedure we trade-off visualization reproducibility against prediction error by plotting the two parametrically as we vary control parameters, say RBF kernel width or the sparsity of the reconstructed pre-image subspace dimension.

3. Experiments

Functional magnetic resonance imaging (fMRI) represents a challenging problem for statistical learning at the interface between neuroscience and cognitive psychology. With the neuroimaging community's strong focus on visualization of predictive models this makes an interesting case for our approach to sparse denoising using kernel PCA.

A fMRI data set was acquired by Dr. Egill Rostrup at Hvidovre Hospital on a 1.5 T Magnetom Vision MR scanner. The scanning sequence was a 2D gradient echo EPI (T2-weighted) with 66 ms echo time and 50° RF flip angle. The images were acquired with a matrix of 128 × 128 pixels, with FOV of 230 mm, and 10 mm slice thickness, in a para-axial orientation parallel to the calcarine sulcus, hence capturing possible activation in visual cortices. The

Fig. 2. Denoising of synthetic data. The left panel shows both activation patterns of the artificial signal, while the top panel shows examples of the synthetic data (5 samples from each state). The two lower panels show the pre-image reconstructions of the sparse and dense methods, respectively.

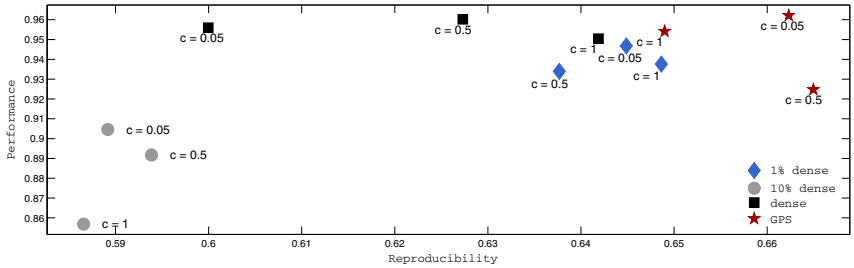


Fig. 3. Prediction/reproducibility plots using all scans for the single slice fMRI visual block activation experiment. The GPS estimate when using a non-linear kernel are seen to outperform all other estimates in terms of combined prediction and reproducibility measures. Location in the upper right corner is preferred.

visual stimulus paradigm consisted of a rest period of 20 s of darkness using a light fixation dot, followed by 10 s of full-field checkerboard reversing at 8 Hz, and ending with 20 s of rest (darkness). In total, 150 images were acquired in 50 s, corresponding to a period of approximately 330 ms per image. The experiment was repeated in 10 separate runs containing 150 images each. In order to reduce saturation effects, the first 29 images were discarded, leaving 121 images for each run. We use a simple on–off activation reference function for supervision of the classifier. The reference function is off-set by 4 s to adjust for the hemodynamic delay.

3.1. Synthetic data

As an initial illustration of the performance of the sparse pre-image reconstruction scheme we construct an artificial data set. Initially we use the non-iterative scheme for source separation introduced by Molgedey and Schuster (1994) for independent component analysis (ICA) in order to achieve a realistic model of the fMRI noise environment. When using 10 components for the decomposition, the signal manifold is contained in the leading three independent components. Hence, we discard the three first components, and reconstruct the projections onto the remaining components, thereby achieving a realistic representation of the noise present in fMRI. We add an artificial signal with different activation patterns for each state to the reconstructed noise, and use this data set as synthetic data. The signal-to-noise ratio of the synthetic data is $\mathrm{SNR} = \frac{1}{N}\sum_{i=1}^{N} \frac{\mu^i_{signal}}{\sigma^i_{noise}} = 2.66$, where μ^i_{signal} is the amplitude of the artificial signal in the i'th scan and σ^i_{noise} is the standard deviation of the background pixels in scan i. Examples of the synthetic data and the artificial signal can be seen in the top and left panel of Fig. 2, respectively.

The synthetic data is randomly split into two equal sized subsets: five runs for training and five runs for testing. The scale parameter of the Gaussian kernel is chosen as the 5th percentile of the mutual distances, while the dimension of the principal subspace is chosen to $q = 5$. We then use the fixed point iteration scheme in Eq. (7) for computing a dense (denoised) pre-image, while the GPS framework is used for finding a sparse pre-image by minimizing Eq. (11). Examples of the denoised scans are shown in the two lower panels of Fig. 2. It is evident that both the dense and sparse reconstructions capture the signal. However, for the sparse pre-image the signal is more distinctive and the reconstructed scans in general suffer from less noise.

3.2. Original data

Similar to the analysis of the synthetic data, we initially split the original data into two equal sized subsets, and choose the scale parameter of the Gaussian kernel as the 5th percentile of the mutual distances. The kPCA basis is defined by the principal components with a significant correlation with the reference function

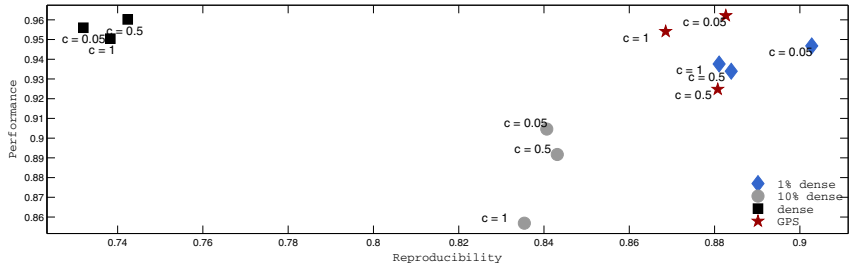


Fig. 4. Prediction/reproducibility plots using only the active scans for the single slice fMRI visual block activation experiment. Compared to Fig. 3, a much higher reproducibility is seen across all methods. Furthermore, the sparse reconstructions clearly outperform the dense estimate.

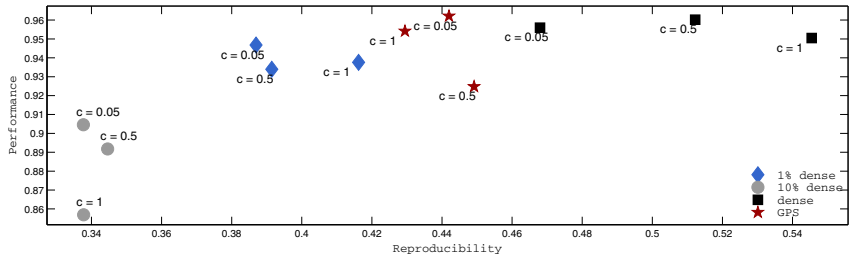


Fig. 5. Prediction/reproducibility plots using only the baseline scans for the single slice fMRI slice visual block activation experiment. It is evident that all methods suffer from lower reproducibility in the baseline scans. Additionally, the dense estimate is seen to be superior for these scans.

($\rho > 0.2$). An animation of both the dense and sparse pre-image reconstructions of the entire test set can be found at http://www2.imm.dtu.dk/~tjab.

In order to assess the quality of the reconstructions we apply a split-half resampling framework for estimating prediction/reproducibility metrics as mentioned in Section 2.2. The dimensionality of the principal subspace q is defined as above. Three values of the smoothing scale of the Gaussian kernel are investigated ranging from a very nonlinear feature space map to a near-linear case equivalent to conventional PCA, viz., the scale is chosen as (1) the 5th percentile, (2) the median and (3) the maximum of the mutual distances in the data set. Furthermore, we run the experiments with a varying degree of sparsity imposed on the reconstruction (points along the GPS path). Additional to the fully dense pre-image and the final GPS reconstruction, we also estimate pre-images which have fixed sparsity, viz. 1% and 10% dense. The converged GPS solution is on average 2.5% dense. For each experiment 45 split-half resamples are performed. The predictive power is measured as the classification rate of the linear classifier trained on the denoised scans. Again the data is split in two equal sized subsets for training and testing respectively. As the test and training data are independent, the classification rate estimate is an unbiased estimator of performance. Reproducibility is measured as the mean correlation between the pre-image from each split-half resample.

The results are summarized in Figs. 3 through 5. The figures show the prediction/reproducibility plot (*pr*-plot) using all scans, only the active scans, and only the baseline scans, respectively. Useful control parameters (smoothing parameter scale, sparsity level) provide high predictive power and reproducibility, c.f., locations in the upper right of the plot are preferred. It is seen how the sparse reconstructions are superior to the dense estimates,

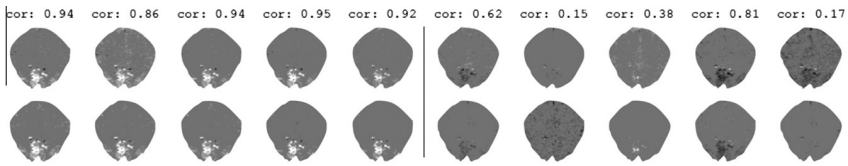


Fig. 6. Example of the GPS reconstruction after projection on the two training sets in a split half experiment (top and bottom panel respectively). The five left panels show denoised active scans, whereas the five panels to the right show baseline scans. The correlation between the two reconstructions are given above each column. The higher reproducibility of the active scans are evident. The extended bright areas in the lower part of the slice in activated scans are located in the primary visual areas.

when evaluating both reliability and predictability for all data. Since the reproducibility was found to differ for the active and baseline scans, the *pr*-plot is shown for these two cases separately in Figs. 4 and 5, respectively. While the best performance for all data with the GPS estimate is a result of a good performance in both active and baseline data, better *pr*-scores are obtained by more sparse (1%) and dense estimators, respectively, for active and baseline scans separately.

In Fig. 6 we show examples of the GPS reconstructions obtained from a particular split, i.e., obtained by denoising based on two non-intersecting data sets. It is evident that the visual activation network estimated is highly reproducible, while the patterns observed in the denoised baseline scans show larger fluctuations distributed widely across brain.

4. Conclusion

In this contribution we addressed the reliability and tuning of sparse pre-image estimation. In a functional brain imaging case study it was shown how sparse reconstruction not only leads to visual appealing pre-images but the estimates are also highly reproducible. We thus recommend to augment the cost function for pre-image estimation in Eq. (5), with a ℓ_1-norm penalty term in order to impose sparsity on the sought pre-image. Using the *pr* evaluation metric we could identify useful values of the two important parameters for sparse denoising kernel PCA, namely the smoothing scale, c, and the sparsity control, λ.

Acknowledgment

The research is funded in part by the Danish Lundbeckfonden through CIMBI Center for Integrated Molecular Brain Imaging.

References

Abrahamsen, T.J., Hansen, L.K., 2010. Regularized pre-image estimation for kernel PCA de-noising input space regularization and sparse reconstruction. J. Signal Process. Systems. doi:10.1007/s11265-010-0515-4.

Arias, P., Randall, G., Sapiro, G., 2007. Connecting the out-of-sample and pre-image problems in kernel methods. In: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 18–23 June.

Bakir, G.H., Weston, J., Schölkopf, B., 2004. Learning to find pre-images. Advances in Neural Information Processing Systems, 16. MIT Press, pp. 449–456.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. 2, 121–167.

Dambreville, S., Rathi, Y., Tannenbaum, A., 2006. Statistical shape analysis using kernel PCA. In: IS&T/SPIE Symp. on Electrical Imaging.

Friedman, J.H., 2008. Fast sparse regression and classification. Technical report, Department of Statistics, Stanford University.

Hansen, L.K., 2007. Multivariate strategies in functional magnetic resonance imaging. Brain Lang. 102 (2), 186–191.

Hansen, L.K., Larsen, J., Nielsen, F.A., Strother, S.C., Rostrup, E., Savoy, R., Lange, N., Sidtis, J., Svarer, C., Paulson, O.B., 1999. Generalizable patterns in neuroimaging: How many principal components? NeuroImage 9 (5), 534–544.

Kwok, J.T.-Y., Tsang, I.W.-H., 2004. The pre-image problem in kernel methods. IEEE Trans. Neural Networks 15 (6), 1517–1525.

Mika, S., Schölkopf, B., Smola, A., Müller, K.R., Scholz, M., Rätsch, G., 1999. Kernel PCA and de-noising in feature spaces. Advances in Neural Information Processing Systems, vol. 11. MIT Press, pp. 536–542.

Molgedey, L., Schuster, H.G., 1994. Separation of a mixture of independent signals using time delayed correlations. Phys. Rev. Lett. 72 (23), 3634–3637.

Mørch, N., Hansen, L., Strother, S., Svarer, C., Rottenberg, D., Lautrup, B., Savoy, R., Paulson, O., 1997. Nonlinear versus linear models in functional neuroimaging: learning curves and generalization crossover. In: Proc. 15th Internat. Conf. on Information Processing in Medical Imaging, 1997, Springer Lecture Notes in Computer Science, vol. 1230, pp. 259–270.

Pereira, F., Mitchell, T., Botvinick, M., 2009. Machine learning classifiers and fmri: A tutorial overview. NeuroImage 45 (1, Supplement 1), S199–S209 (Mathematics in Brain Imaging).

Rice, J.R., White, J.S., 1964. Norms for smoothing and estimation. SIAM Rev. 6, 243–256.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.

Schölkopf, B., Smola, A., Müller, K.-R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10 (5), 1299–1319.

Strother, S., Anderson, J., Hansen, L., Kjems, U., Kustra, R., Sidtis, J., Frutiger, S., Muley, S., LaConte, S., Rottenberg, D., 2002. The quantitative evaluation of functional neuroimaging experiments: The NPAIRS data analysis framework. NeuroImage 15 (4), 747–771.

Thirion, B., Faugeras, O., 2003. Dynamical components analysis of fMRI data through kernel pca. Neuroimage 20 (1), 34–49.

Tibshirani, R., 1994. Regression shrinkage and selection via the Lasso. J. Roy. Statist. Soc. Ser. B 58, 267–288.

Yourganov, G., Chen, X., Lukic, A.S., Grady, C.L., Small, S.L., Wernick, M.N., Strother, S.C., 2011. Dimensionality estimation for optimal detection of functional networks in bold fmri data. NeuroImage 56 (2), 531–543.

Appendix C

Nonlinear Denoising and Analysis of Neuroimages with Kernel Principal Component Analysis and Pre-image Estimation

Peter Mondrup Rasmussen, Trine Julie Abrahamsen, Kristoffer Hougaard Madsen, and Lars Kai Hansen. Nonlinear Denoising and Analysis of Neuroimages with Kernel Principal Component Analysis and Pre-image Estimation. *NeuroImage*, 60(3):1807 – 1818, 2012.
doi:10.1016/j.neuroimage.2012.01.096.

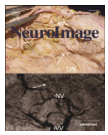
Citations
This paper has been cited by [SCB13, FWXT13].

NeuroImage 60 (2012) 1807–1818

Contents lists available at SciVerse ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg



Technical Note

Nonlinear denoising and analysis of neuroimages with kernel principal component analysis and pre-image estimation

Peter Mondrup Rasmussen [a,b,*], Trine Julie Abrahamsen [a], Kristoffer Hougaard Madsen [a,c], Lars Kai Hansen [a]

[a] DTU Informatics, Technical University of Denmark, Denmark
[b] The Danish National Research Foundation's Center for Functionally Integrative Neuroscience, Aarhus University Hospital, Denmark
[c] Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital Hvidovre, Denmark

ARTICLE INFO

Article history:
Received 5 September 2011
Revised 15 January 2012
Accepted 18 January 2012
Available online 28 January 2012

Keywords:
Multivariate analysis
Classification
Decoding
Nonlinear modeling
Kernel PCA
Pre-image estimation
NPAIRS resampling

ABSTRACT

We investigate the use of kernel principal component analysis (PCA) and the inverse problem known as *pre-image* estimation in neuroimaging: i) We explore kernel PCA and pre-image estimation as a means for image denoising as part of the image preprocessing pipeline. Evaluation of the denoising procedure is performed within a data-driven split-half evaluation framework. ii) We introduce manifold navigation for exploration of a nonlinear data manifold, and illustrate how pre-image estimation can be used to generate brain maps in the continuum between experimentally defined brain states/classes. We base these illustrations on two fMRI BOLD data sets – one from a simple finger tapping experiment and the other from an experiment on object recognition in the ventral temporal lobe.

Introduction

Analysis of neuroimaging data sets is challenging. Typical data sets are characterized not only by variation due to manipulation of an experimental variable of interest. A significant amount of variation origins from, e.g., between subject variability, subject movement, physiological noise and scanner drift. One can consider brain scans as measurement points residing in a high dimensional space (thousands of voxels), where the underlying structure of the signal may be characterized by a low-dimensional structure embedded in the high dimensional space. Such underlying signal structure may by quite complex. Consider a neuroimaging experiments with two experimental conditions *A* and *B*. The observation points may not only be present as cluster structures around class centroids characterizing A and B in voxel space. We could imagine, that the signal structure may follow trajectories induced by e.g. learning effects or temporal structure in physiological noise or scanner drift. Note that also signal components considered as noise/artifacts may be structured as well as highly reproducible. Our primary aim is apply flexible denoising methods that are capable in modeling such complex underlying signal structure. Additionally, we impose a constraint on the structure modeling, and aim for denoising procedures that allow multivariate models linking brain scans to behavior to provide both good prediction performance as well as a stable/reproducible visualization.

The diversity and complexity of the acquired signals has lead to the development of a wide range of preprocessing and analysis strategies. A widely used model of the signal is the general linear model (GLM) (Friston et al., 1994) linking the observed brain signals and knowledge on experimental settings and known nuisance effect as modeled in the design matrix. Within the GLM modeling framework we can construct a version of the data with less noise, by projecting the data onto the part of the data space that is orthogonal to the subspace spanned by the nuisance regressors, e.g. (Glover et al., 2000).

Different decomposition methods have been introduced in order to separate the signal of interest from noise components (hereafter referred to as signal and noise) prior to data analysis. Principal component analysis (PCA) defines the data in terms of a new basis set composed by a series of orthogonal eigenimages (Bullmore et al., 1996; Hansen et al., 1999; Thomas et al., 2002). Image denoising is performed by partitioning the basis set into a signal set and a noise set and projecting the data onto the linear subspace in the voxel space spanned by the signal basis set. Independent component analysis (ICA) attempts to identify spatially or temporally statistically independent sources of variation (McKeown et al., 1998). Denoising is performed by either retaining components identified as signal components or projecting out noise components from the data set (Thomas et al., 2002; Tohka et al., 2008).

* Corresponding author at: DTU Informatics, Technical University of Denmark Richard Petersens Plads, Kongens Lyngby, Denmark.
E-mail address: pmra@imm.dtu.dk (P.M. Rasmussen).

1053-8119/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.neuroimage.2012.01.096

For GLM, PCA, and ICA we can consider the process of denoising mathematically as the following two step procedure. Consider a data set X $(D\times N)$ holding N scans with D voxels in the columns. i) First the scan to be denoised x $(D\times 1)$ is projected onto a linear subspace $s = Wx$ on dimension k where the rows of the $(k\times D)$ matrix W spans the subspace. E.g. in the GLM approach the rows will span a vector space that is orthogonal to nuisance effect as modeled in the design matrix, while in PCA the rows will be a subset of the orthogonal basis images. ii) Hereafter s is reconstructed in voxel space by the inverse transformation $\hat{x} = W^{-}s$, where $(\cdot)^{-}$ denote the Moore–Penrose pseudo inverse.

The description above highlights that GLM, PCA and ICA are all linear methods. The subsequent denoising relies on the assumption that the relevant data structure resides on a linear manifold embedded within the high-dimensional voxel space. However, since signal structures in the acquired data may exhibit nonlinear properties, linear models may not always be appropriate (Thomas et al., 2002). PCA can be extended to a nonlinear feature extractor through the kernel methodology (Mika et al., 1999; Schlökopf et al., 2000).

Kernel PCA is a nonlinear generalization of PCA, in which the basic idea is to map the data from voxel space (hereafter also referred to as input space) to a reproducing kernel Hilbert space (RKHS), referred to as feature space, and then perform PCA on the mapped data. Often, the feature space is high (or even infinite) dimensional and kernel PCA is implemented by exploiting the *kernel trick*, e.g. Schölkopf et al. (1998a). By use of the kernel trick, calculations can be performed implicitly in feature space as long as the data only appear as inner-products in the model formulation. Hence, we can easily perform the above step i) through the kernel methodology. The main challenge in denoising by kernel PCA is the mapping of denoised feature space points back into input space — also known to as the *pre-image problem*, step ii) above. Since the feature space mapping is not bijective for many choices of the kernel function, pre-image estimation is inherently ill-posed.

Even though, the practical use of nonlinear kernel based preprocessing methods has been limited, recent years have seen an increased interest in applying kernel PCA as a preprocessing and analysis tool in the field of neuroimaging. Kernel PCA has been applied as a preprocessing/feature extraction stage in a computer-aided diagnosis system, that was build to distinguish Alzheimer's disease subjects from a control group (López et al., 2009). These authors used kernel PCA to extract nonlinear features from single-photon emission tomography images and subsequently trained linear and nonlinear classifiers on the kernel PCA feature representation. Within analysis of fMRI, Thirion and Faugeras (2003) used kernel PCA to perform nonlinear dimensionality reduction prior to modeling, while Song et al. (2008) used kernel PCA and pre-image estimation to derive a nonlinear frequency analysis scheme for noise removal. Guo (2010) used kernel PCA, based on a multi-subject kernel matrix, as a feature extraction step in a predictive modeling framework.

The contributions of the present study are: i) We investigate the performance of kernel PCA and the subsequent pre-image estimation as a tool for noise reduction in fMRI. The evaluation is performed within the nonparametric, prediction, activation, influence, reproducibility, re-sampling (NPAIRS) framework (Strother et al., 2002), a data-driven split-half evaluation framework in which we build multivariate models of the data and base the evaluation on both brain state predictability and the reproducibility of brain maps extracted from multivariate models. ii) We introduce *manifold navigation* for exploration of a nonlinear data manifold, and illustrate how pre-image estimation can be used to generation brain maps in the continuum between experimentally defined brain states/classes. Our procedure extends the hyperplane navigation procedure proposed for linear models by Sato et al. (2008).

The remainder of this paper is organized as follows: In Materials and methods section we present the two fMRI data sets used for illustration, review the basic concepts of kernel PCA and pre-image estimation and provide a description of the modeling and resampling procedures. Results section presents the results, which is then discussed in the Discussion section. Finally, Conclusion section concludes the paper.

Materials and methods

To illustrate the use of kernel PCA and subsequent pre-image reconstruction we present two application examples. *Illustration I* focuses on image denoising in neuroimaging data sets. *Illustration II* presents a procedure that attempts to explore a low dimensional manifold embedding by construction of activation patterns in input data space.

Data sets

Finger tapping data set

Finger tapping consisted of two paced motor conditions in the following sequence: (RIGHT) right hand finger tapping, (LEFT) left hand finger tapping. Pacing was provided by means of a red (LEFT condition) or green (RIGHT condition) circle flashing at 1 Hz presented at the center of a screen. Each condition was presented for 20 s followed by 9.88 s of rest with no finger tapping. The stimulation cycle was repeated 10 times in the experimental run, and 240 scan volumes were acquired in total. One experimental run per subject was conducted. Data from 28 healthy subjects were used in the analysis. They gave informed consent as approved by the local Ethics Committee. Further details on the experiment and data acquisition are found in Rasmussen et al. (2012).

Preprocessing of the fMRI time series was conducted using a default strategy in the SPM8 software package (http://www.fil.ion.ucl.ac.uk/spm) and comprised the following steps: (1) Rigid body realignment of echo planar imaging (EPI) images to the mean image in the time series, (2) co-registration to the magnetization prepared rapid acquisition gradient echo (MPRAGE) image, (3) spatial normalization of MPRAGE images to the MNI152 template (Montreal Neurological Institute template), (4) re-slicing of EPI images into MNI space at 3 mm isotropic voxels using the estimated normalization warp fields, (5) spatial smoothing of spatial normalized EPI images using an isotropic Gaussian filter (6 mm FWHM). Low frequency components were removed with a set of discrete cosine basis functions up to a cut-off period of 128 s. The mean rest-condition volume was subtracted from each subject, based on the last two images of each rest period. Finally, the data were masked with a rough whole-brain mask (57998 voxels).

Object recognition data set

This data set originates from the experiment of Haxby et al. (2001) on face and object representation in the human ventral temporal cortex.[1] The data set consists of six subjects with 12 experimental runs per subjects. In each run the subjects were viewing gray scale images of eight object categories {bottle, cat, chair, face, house, scissors, scrambled, shoe} grouped into 24 s blocks separated by rest periods. Further details on the experiment and data acquisition are found in Haxby et al. (2001).

The data were preprocessed primarily with FSL utilities (Smith et al., 2004) and comprised the following steps: (1) The EPI images were scull-stripped with BET, (2) correction for rigid-body movement with MCFLIRT, (3) the time series were linearly de-trended and standardized within each run. Voxels that entered further analysis were

[1] The data were obtained from the PyMVPA web site http://www.pymvpa.org. The authors of Haxby et al. (2001) hold the copyright of the dataset and it is available under the terms of the Creative Commons Attribution-Share Alike 3.0 license.

identified based on subject specific masks (`mask_vt.nii`) provided with the data set comprising between 307 and 675 voxels.

Kernel PCA

We are interested in identifying a given signal manifold using nonlinear manifold learning. Denoising is achieved by projecting noise distorted data onto this clean manifold. Denoising by kernel PCA can be thought of in three steps. First we map the input space data to a feature space using nonlinear maps. In feature space the signal manifold is expected to be linear, and linear PCA is employed to identify the manifold. In the final step the pre-images of the projected points in the PCA subspace of the feature space are reconstructed in the original input space. Unfortunately, finding a reliable pre-image is entirely non-trivial due to the ill-posed nature of many choices of feature space maps, and hence several algorithms have been suggested, see e.g. Mika et al. (1999), Kwok and Tsang (2004), Dambreville et al. (2006).

In the following we will briefly summarize kernel PCA as introduced by Schölkopf et al. (1998a). Let $\varphi : \mathcal{X} \mapsto \mathcal{F}$ be a possible nonlinear map from the D-dimensional input space, $\mathcal{X}$, to the high dimensional feature space, $\mathcal{F}$. Now, let $\{x_1,\ldots,x_N\}$ be N data points in $\mathcal{X}$ and $\{\varphi(x_1),\ldots,\varphi(x_N)\}$ be the corresponding images in $\mathcal{F}$. In order to apply the kernel trick and thereby avoid explicit calculations in $\mathcal{F}$, the kernel function $k(x,x') = \varphi(x)^\top \varphi(x')$ is introduced (see notation[2]).

Kernel PCA can then be performed by solving the eigenvalue problem $\tilde{K}\alpha_i = N\lambda_i\alpha_i$, where $\tilde{K}$ is the centered kernel matrix. The projection of a centered feature space map, $\tilde{\varphi}(x)$, onto the *i'th* principal component in $\mathcal{F}$ is given by $\beta_i = \sum_{n=1}^{N} \alpha_{in}\tilde{k}(x,x_n)$, where the α_i's have been normalized.

Furthermore, the projection of $\varphi(x)$ onto the subspace spanned by the first q eigenvectors can be found as $P_q\varphi(x) = \sum_{i=1}^{q}\beta_i\sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(x_n) + \bar{\varphi} = \sum_{n=1}^{N}\tilde{\gamma}_n\tilde{\varphi}(x_n) + \bar{\varphi}$, where $\bar{\varphi}$ is the mean of the φ-mapped data and $\tilde{\gamma}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$. We expect the signal manifold to be described by the leading q principal components, therefore only retaining $q<N$ components when performing kernel PCA denoising.

Equivalent to linear PCA, the squared reconstruction error is minimal and the retained variance is maximal for kernel PCA. However, these properties hold in $\mathcal{F}$ not in $\mathcal{X}$. For a more thorough derivation of kernel PCA and details on e.g. kernel centering the reader is referred to Schölkopf et al. (1998a). In the following we focus on the Gaussian kernel of the form $k(x,x') = exp(-\frac{1}{\sigma}||x-x'||^2)$, where σ controls the width of the kernel. The linearity of the kernel embedding is proportional to σ and in the limiting case, as $\sigma\to\infty$, kernel PCA approaches linear PCA as discussed in Appendix B.

Pre-image estimation

As previously described, the final step in denoising by kernel PCA consists of applying the inverse map of φ to reconstruct a data point in input space from a point in feature space. Assuming that the given feature space point lies in the span of $\{\varphi(x_i)\}_{i=1}^{N}$ implies that it can be represented as a linear combination of the training images, i.e., $P_N\varphi(x)$, without loss of generality. Thus, the pre-image problem consists of finding a point $z\in\mathcal{X}$ such that $\varphi(z) = P_q\varphi(x)$. z is then called the pre-image of $P_q\varphi(x)$.

Since a function has an inverse if and only if it is bijective, φ will not be invertible for most nonlinear kernel functions, and thus the pre-image problem is ill-posed (Burges, 1998; Schölkopf et al., 1998b; Mika et al., 1999; Schölkopf et al., 1999; Kwok and Tsang, 2004; Dambreville et al., 2006; Arias et al., 2007). For many choices of kernel $\dim(\mathcal{F}) \gg \dim(\mathcal{X})$, and it follows that not all points in $\mathcal{F}$ or even the subspace spanned by $\{\varphi(x_i)\}_{i=1}^{N}$ is the image of any $x\in\mathcal{X}$. Furthermore, whenever φ is not injective, uniqueness of a recovered pre-image is not guaranteed.

Since an exact pre-image often does not exist, various approaches to the nonlinear optimization problem of finding an approximate pre-image have been developed in recent years (Mika et al., 1999; Kwok and Tsang, 2004; Dambreville et al., 2006). The original work by Mika et al. (1999) proposed a fixed-point iterative approach by seeking a point in input space which maps into a point in feature space 'as close as possible' to $P_q\varphi(x)$ (see Fig. 1). Thus, the pre-image estimate is defined as a point which minimizes the Euclidean distance between $\varphi(z)$ and $P_q\varphi(x)$ with respect to z. Straightforward manipulations allow simplification of this quadratic objective function

$$R(z) = ||\varphi(z) - P_q\varphi(x)||^2 = k(z,z) - 2\sum_{n=1}^{N}\gamma_n k(z,x_n) + \Omega. \tag{1}$$

where z-independent terms are collected in Ω, and $\gamma_n = \tilde{\gamma}_n + \frac{1}{N}\left(1 - \sum_{j=1}^{N}\tilde{\gamma}_j\right)$ as accounted for in Appendix C.

The fact that the minima of Eq. (1) are among points for which $\delta R/\delta z = 0$, leads to the following fixed-point iteration for the Gaussian kernel:

$$z_{t+1} = \frac{\sum_{n=1}^{N}\gamma_n exp(-||z_t - x_n||^2/\sigma)x_n}{\sum_{n=1}^{N}\gamma_n exp(-||z_t - x_n||^2/\sigma)} \tag{2}$$

As any other iterative approach to nonlinear optimization problems, the method of Mika et al. (1999) can suffer from convergence to local minima and sensitivity to the initialization. Abrahamsen and Hansen (2009) suggested stabilizing the pre-image estimate by input space regularization of the objective function in Eq. (1). In such cases the chosen nonlinearity of the Gaussian kernel will affect observation points' feature space projections, whereas the regularization will affect the pre-image estimation. Abrahamsen and Hansen (2011) investigated the pre-image variability as a function of the nonlinearity of the kernel function. It was observed that the stability of the pre-image estimate is strongly related to the nonlinearity of the feature space map. We here use un-regularized pre-image estimates as the kernels applied are not highly nonlinear.

While Mika et al. (1999) introduced the iterative approach shown above, both Kwok and Tsang (2004) and Dambreville et al. (2006) proposed closed-form solutions. For comparison we also provide results using the method of Kwok and Tsang (2004), which is not based on minimizing Eq. (1) but on the assumption that for any two observations x_i and x_j there exists a simple relation between their Euclidean distance in input space and the distance between the corresponding φ-mapped images in feature space. The relation between the distance measures is obtained by exploiting the idea of multidimensional scaling, where a low dimensional distance preserving manifold is sought. Instead of using all the training points, only the k nearest neighbors in feature space are used for the pre-image estimation. The basic idea of Kwok and Tsang's method is to estimate the pre-image by projection onto the subspace in input space spanned by the chosen neighbors.

Image denoising with kernel PCA

For a particular parameter combination of the kernel width, σ, of the Gaussian kernel and the dimensionality of the kernel PCA subspace, q, a denoised version, $Z^{(\sigma,q)}$, of the original data observations X was determined. In both data sets we performed image denoising at the subject level. A kernel PCA basis was estimated from all scans

[2] Bold uppercase letters denote matrices, bold lowercase letters represent column vectors, and non-bold letters denote scalars. a_j denotes the *j'th* column of A, while a_{ij} denotes the scalar in the *i'th* row and *j'th* column of A. Finally, 1_{NN} is a $N\times N$ matrix of ones.

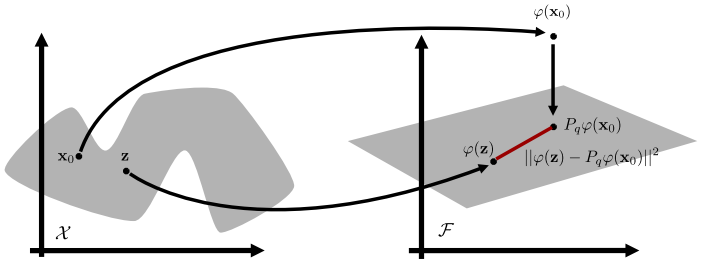


Fig. 1. The pre-image problem in kernel PCA denoising concerns estimating z from x_0, through the projection of the image onto the principal subspace in feature space, $\mathcal{F}$.

of a particular subject, and the images were subsequently denoised by projecting the images onto the kernel PCA basis followed by pre-image estimation. For the finger tapping data set we created denoised versions of the original data set by varying the parameters over the grid $\sigma \in [2^{-3}, 2^{-2}, \ldots, 2^{10}]$ and $q \in [2, 4, 8, 16, 32, 50, 75, \ldots, 175, 240]$. For the object recognition data set we explored the grid $\sigma \in [2^{-3}, 2^{-2}, \ldots, 2^{10}]$ and $q \in [10, 20, \ldots, 250]$. The kernel width was scaled relative to the average input-space distance measure of the Gaussian kernel to the nearest 25% points across all data points. For Kwok and Tsang's reconstruction method we initially considered $k = \{5, 10, 15, 20, 50\}$ neighbors in the finger tapping data set and found no major impact on the model performance. For the results reported on the finger tapping data set we use ten neighbors as suggested in Kwok and Tsang (2004). In the object recognition data set we found that a relatively large number of nearest neighbors was required to achieve stable model performance (see the Supplementary materials Fig. 1). Hence, we report results based on 500 nearest neighbors for the object recognition data set. A relative change below 10^{-9} was used as a convergence criterion for Mika's method.

Illustration I — image denoising

Evaluation of the effect of image denoising was performed by means of multivariate classification analysis. In the following we outline the classification setup and the model evaluation scheme.

Classification analysis

In the two data sets we defined classification tasks as follows. In the finger tapping data set we performed a 'whole brain' and single scan classification and formulated a binary classification tasks (LEFT vs. RIGHT). For the classification analysis we extracted scans from the (LEFT) and (RIGHT) epochs, discarding two transition scans at the start of each block giving 120 scans per subject for the analysis. In the analysis of the object recognition data set we formulated an eight class single scan classification task based on partition of scans into the eight object categories. Nine scans were extracted from each stimuli block giving 864 scans per subject for the analysis (only 792 scans available for subject number five).

For classification we used Fisher's linear discriminant analysis (FDA) which is a well known method that considers dimensionality reduction and classification jointly. FDA seeks to find an optimal subspace where the projected class means are separated the most in terms of variance, see e.g. Hastie et al. (1995, 2009). Identification of the FDA subspace involves the between-class scatter matrix and the pooled-scatter matrix, and requires the pooled-scatter matrix to be nonsingular. Often in neuroimaging data sets the scan dimensionality (number of voxels) exceeds the number of scans, hence regularization is required. We here used a method also referred to as regularized/penalized discriminant analysis (Hastie et al., 1995), where the identity matrix scaled by a regularization parameter λ is added to the pooled scatter matrix. For model fit we used the algorithm developed by Zhang et al. (2009b). In a C-class classification problem the FDA subspace is spanned by C-1 basis vector, also referred to as canonical variates. On top of the FDA basis we implemented a simple nearest mean classifier to evaluate the prediction accuracy. To interpret/visualize the FDA model we use two strategies. i) Use the canonical variates to obtain C-1 maps that reflect important directions in the voxel space. ii) Derive a sensitivity map for the classifier as suggested by Kjems et al. (2002). The sensitivity map resides in voxel-space, and each value in the sensitivity map reflects the relative importance of a particular voxel to the classifier outputs. In the case of a two-class classification setup the sensitivity map is identical to the squared elements of the (single) canonical variate. We refer to Kjems et al. (2002) for further details on the sensitivity map model visualization procedure. In the following we will also refer to the canonical variates/sensitivity maps as 'weight vectors'.

Resampling and model evaluation

In the following we provide a general description of the model evaluation procedure. This is followed by a detailed description of the specific implementations of the model evaluations for the two data sets.

NPAIRS resampling framework. Model evaluation was based on the nonparametric, prediction, activation, influence, reproducibility, resampling (NPAIRS) scheme (Strother et al., 2002, 2010). In this split sampling framework the data observations were split into two equal partitions. The classification model was trained on the first split and the prediction accuracy was estimated from the second split and vice versa, yielding two estimates of the prediction accuracy (we here report % accuracy/100). These prediction accuracies were averaged and considered as the prediction metric *(p)* of the NPAIRS scheme. In addition, the Pearson's correlation coefficient between model weight vectors (canonical variates/sensitivity maps) derived from the two models was calculated as the spatial reproducibility metric *(r)*. By plotting the *p* metric vs. the *r* metric and varying the regularization parameter λ in FDA a *pr-curve* was constructed. This curve reflects a trade-off between prediction accuracy and pattern reproducibility. To derive a brain map based on the weight maps of the two models we applied the reproducible statistical parametric image (rSPI) mapping procedure of the NPAIRS framework (Strother et al., 2002). The rSPI mapping procedure provides an assessment of the similarity between the weight vectors derived from the two models trained on independent splits of the data. Specifically, the mapping procedure proceeds as follows: i) Each weight vector is scaled to unit standard deviation. (Note, that we here do not perform demeaning — only scaling) ii) The two weight vectors are plotted against each other and the points form a scatter cloud. iii) The scatter

cloud is projected onto a signal axis and an orthogonal noise axis. iv) The projection onto the signal axis (sum of split-half weight vectors) is scaled by the standard deviation of the noise projection (difference of split-half weight vectors), which gives a reproducible volume (rSPI). The rSPI can be interpreted as a Z-score pattern under the assumption that the noise distribution (the projection of the scatter cloud onto the noise axis) is Gaussian (Strother et al., 2002). Under this assumption we let rSPI(Z) denote the reproducible volume. The rSPI(Z) can be compared to the theoretical $\mathcal{N}(0,1)$ distribution for further analysis, e.g. statistical thresholding. Note that the rSPI(Z) differ from the conventional t-maps derived from univariate analyses. While t-map reflects to what extend a voxel behaves according to a prescribed design matrix, the rSPI(Z) reflects similarities in maps derived from models trained on independent splits of the data. An algorithm for the NPAIRS scheme and rSPI mapping procedure is provided in Appendix **D**.

As a model performance metric we used the minimum Euclidean distance between the pr-curve and the point (1,1) in the pr-plot. Hence, we focus not only on maximization of prediction accuracy. We also aim for a stable visualization and prefer classification models with a high degree of reproducibility of the spatial patterns extracted from the models. It has been demonstrated in a series of studies that in terms of signal detection it is relevant also to consider reproducibility as a performance metrics in the model optimization, see e.g. Strother et al. (2002, 2004), Yourganov et al. (2011), Rasmussen et al. (2012). The Euclidean distance to (1,1) in the pr-plot has previously been used as a means for evaluation of preprocessing strategies, see e.g. Zhang et al. (2009a) and references therein.

Finger tapping data set. We split the finger tapping data set into a training set of 10 subjects and a test set of 18 subjects. Selection of the denoising parameters (σ and q) was based on the training set. The training set was repeatedly split into two disjoint sets, each with five subjects, and model performance was evaluated using the NPAIRS resampling scheme. To evaluate the reproducibility we used the canonical variate. 20 NPAIRS resampling splits were performed, and the average minimum distance on the pr-curve to the point (1,1) was obtained across the entire parameter grid. The test set was then denoised using the parameter combination giving the minimum distance. The impact of image denoising was then evaluated by constructing pr-curves based on analysis of the raw test data and denoised test data within the NPAIRS resampling framework. 20 NPAIRS splits were performed, where nine subjects were randomly assigned to each of the split halves.

Object recognition data set. In the object recognition data set we performed the evaluation of image denoising at the subject level. For a particular subject the data was split into a training and a test set — each with six runs. As with the finger tapping data set the selection of denoising parameters was based on training set. The training set was repeatedly split into two disjoint sets, each with three runs, and model performance was evaluated using the NPAIRS resampling scheme. With eight classes we obtain seven canonical variates. To evaluate the reproducibility we considered the first canonical variate (Chen et al., 2006). When training FDA models on different data samples the canonical variates of the FDA models are defined up to a sign and permutation ambiguity. To align canonical variates across splits we used the reference set filtering described in Strother et al. (2002). In the reference filtering procedure we initially fit a model to the entire training set and extract a set of canonical variates from this model. This set is considered as a reference set. When performing the resampling splits, we then permute and flip signs of the split's individual canonical variates in order to maximize the correlation with the reference set. 10 NPAIRS resampling splits were performed (all possible combinations of runs), and the average minimum distance on the pr-curves to the point (1,1) was obtained across the entire parameter grid. Denoising parameters were then selected according to minimization of the distance to (1,1) metric. The impact of image denoising was then evaluated by constructing pr-curves based on analysis of the raw test data and denoised test data (six runs) within the NPAIRS resampling framework. The entire evaluation procedure was repeated 10 times, with different runs randomly assigned to the training and test sets in each repetition.

Illustration II — manifold navigation

Our second illustration of use of pre-images focuses on exploration of the underlying data structure. Here our concern is to perform the inverse mapping of points residing on the embedding manifold back to the input space data. This approach is inspired by the hyperplane navigation procedure introduced for linear models (Sato et al., 2008) and the morphing/transition method for kernel PCA followed by pre-image estimation (Kwok and Tsang, 2004; Kim et al., 2005).

For illustration we focus on characterizing the embedding manifold identified by kernel PCA by exploring the continuum of points on the line connection class means in the feature space and constructing corresponding brain maps (pre-images). Hence, our objective is to perform an interpolation or prediction of activation maps. The mapping procedure, that we here name manifold navigation proceeds as follows: i) Define the underlying structure of the data in terms of kernel PCA governed by the parameters k and σ. To simplify, we here use the kernel PCA parameters k and σ identified by the resampling procedure in Illustration I, ii) obtain feature space projections of class means and multiple deviations along the line joining the class means, iii) define points of interest in feature space in terms of class means and standard deviations, iv) perform pre-image estimation to map interpolated points into the input data space. In the finger tapping data set we acquired points on the line connection the left- and right centroids, while we considered the bottle-, scissors-, and shoe centroids in the object recognition data set.

In summary, the aim of the manifold navigation procedure is to construct input data space patterns that reflect localized positions on the embedding manifold.

Results

Illustration I — image denoising in the finger tapping data set

Fig. 2 shows the evaluation results of the effect of image denoising in the finger tapping data set. Image denoising was based on Mika's method. Fig. 2(A) shows model performance, as measured by the minimum distance from the pr-curve to the point (1,1), based on the 10 subjects in the training data set. The distance first decreases with an increased number of retained components in the kernel PCA subspace and then tends to increase with at a high number of components retained. Least distances are observed with 16–32 components retained. For a fixed number of components there is a general tendency to decreased distance with increasing width of the Gaussian kernel. Fig. 2(B) shows pr-curves based on analysis of the 18 subjects in the test data set. In general we observe high accuracies and reproducibilities, and the models used on denoised data are characterized by an increased reproducibility compared to models build on the raw data. For the raw data the minimum distance was 0.100 and the corresponding prediction accuracy and reproducibility was 0.994 and 0.900 respectively. For the denoised data based on the Mika's method the minimum distance was 0.0896 and the corresponding prediction accuracy and reproducibility was 0.994 and 0.911 respectively, thus maintaining the prediction accuracy and increasing the reproducibility in comparison to the models build on the raw data set. Denoising did not result in increased prediction accuracy ($p = 0.45$), while the denoising lead to a significant increase in

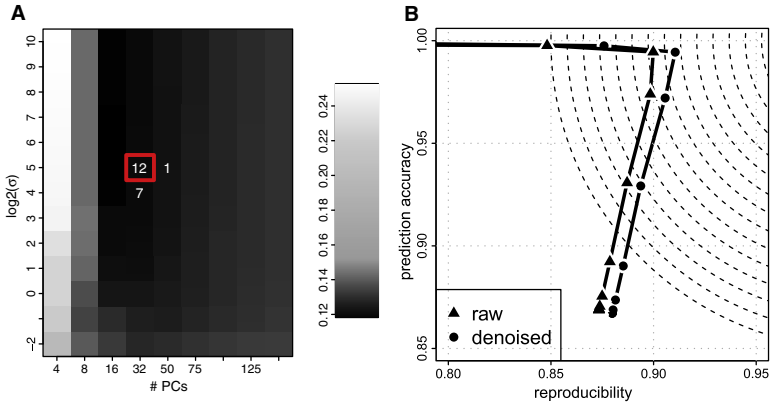


Fig. 2. Effect of denoising in the finger tapping data set with Mika's estimation method. Panel (A); Model performance across part of the explored parameter grid (kernel width and kernel PCA subspace dimensionality) based on 10 subjects. Denoising was performed at the subject level. The model performance was measured as the minimum distance on the pr-curve to the point (1,1). Resampling was performed within the NPAIRS resampling framework (see the Resampling & model evaluation section). The grid shows the average distance across 20 NPAIRS resampling splits. The white numbers indicate the frequency at which a particular parameter combination had the lowest distance on the pr-curve to the point (1,1) across the splits. Panel (B); Model performance based on denoised and raw data from 18 subjects (different from subjects used in Panel (A)). Denoising parameters were selected according to the red square in Panel (A). The pr-curves show model performance in terms of prediction accuracy and pattern reproducibility, where the pr-curves are traced out by varying the regularization parameter in the Fisher's discriminant analysis classification model. The isolines indicate distances to the point (1,1). Denoising did not result in increased prediction accuracy ($p = 0.45$), while the denoising lead to a significant increase in reproducibility $p < 0.001$ (nonparametric permutation test).

reproducibility ($p < 0.001$) as assessed with a nonparametric permutation test.

For the denoised data based on Kwok and Tsang's method the minimum distance was 0.0942 and the corresponding prediction accuracy and reproducibility was 0.987 and 0.910 respectively, hence a decrease in prediction accuracy ($p = 0.001$) and an increased reproducibility ($p < 0.001$) relative to the raw data set. Details on the permutation test are provided in the Supplementary materials.

In the following we present brain maps based on images denoised with Mika's method that showed the best performance in terms of the pr-distance metric. Fig. 3 shows the effect of image denoising on spatial brain maps. Fig. 3(A) is based on the FDA classification models trained within the NPAIRS framework. The maps were thresholded according correction for multiple comparisons by the false discovery rate (FDR) procedure (Benjamini and Hochberg, 1995) using the theoretical $\mathcal{N}(0, 1)$ distribution to obtain p-values for the average reproducible statistical parametric images ($\overline{rSPI(Z)}$s). Cerebellar regions (slice −47 to −11), subcortical regions (slice 1), secondary supplementary motor area (S2) (slice 13) and sensorimotor cortex (SMC) and supplementary motor areas (SMA) (slices 37–61) are consistently identified as important by models build on both raw and denoised data. In general we observe highest Z-scores in the $\overline{rSPI(Z)}$ based on models build on the denoised data. At edges of the superthreshold regions, primarily in cerebellum, we observe a small decrease in Z-score values of the $\overline{rSPI(Z)}$. The intersection mask between the FDR thresholded maps comprised 7291 voxels. In the intersection mask 6658 voxels showed an increase in the $\overline{rSPI(Z)}$ value due to image denoising. Additionally, 701 and 91 voxels were uniquely identified in the maps corresponding to denoised and raw data respectively. Fig. 3(B) shows that these unique voxels mainly appear on the edges of the intersection mask.

Illustration I — image denoising in the object recognition data set

Fig. 4 depicts model performance measured in terms of minimum distance from the pr-curve to the point (1,1) across the denoising parameter grid for image denoising with Mika's method for a single subject. In general we observe a preference towards a relatively low

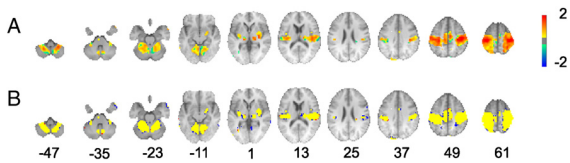


Fig. 3. Spatial maps showing the effect of denoising in the finger tapping data set. The brain maps are based on classification analysis by the Fisher's discriminant analysis performed within the NPAIRS resampling framework. Panel (A); Average reproducible statistical parametric images ($\overline{rSPI(Z)}$s) from models build on raw and denoised data were thresholded according to $p < 0.05$ FDR correction for multiple comparisons. Voxels shown are in the intersection mask of the two thresholded $\overline{rSPI(Z)}$s. Voxel coloring indicates sign and magnitude of the difference between the absolute value $\overline{rSPI(Z)}$s. Warm colors correspond to higher Z-scores in the map based on denoised data, and cold colors correspond to higher Z-scores in the map based on raw data. Panel (B); Binary masks showing voxels surviving thresholding according to FDR correction. Color coding: yellow is an intersection mask (same voxels as in panel (A)), blue is unique to the $\overline{rSPI(Z)}$ based on denoised data, and red is unique to the $\overline{rSPI(Z)}$ based on the raw data. Numbers under the slices denote z coordinates in MNI space. Slices are displayed according to neurological convention (slice left = brain left).

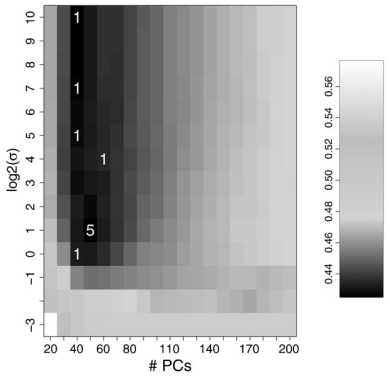


Fig. 4. Effect of denoising in the object recognition data set — impact of denoising parameters. Denoising was performed with Mika's method at the subject level for each combination of the kernel width and the number of principal components. For each parameter combination an evaluation of the impact of image denoising was performed within the NPAIRS resampling framework, and the distances between the pr-maximizing point on the pr-curve to the point (1,1) was measured and used as a model performance metric (see the Resampling and model evaluation section). The distance metric was based on prediction accuracy and reproducibility of the first canonical variate in the FDA model. Selection of denoising parameters was based on six randomly selected runs. The remaining six runs served as a test set for the evaluation of denoising in Fig. 5. The plot shows the average distance metric across 10 resampling iterations (with 10 nested NPAIRS resampling splits within each iteration). The white numbers indicate the frequency at which a particular parameter combination had the lowest distance on the pr-curve to the point (1,1) across the 10 resampling iterations.

number of retained components in the kernel PCA subspace. In general, there is an interaction between the number of principal components and the width of the Gaussian kernel because decreasing the kernel width causes a more flexible model. For a fixed number of principal components the performance tend first to decrease with the width of the Gaussian kernel and again slightly increase at large kernel widths, suggesting that the underlying signal manifold may be nonlinear. In general the maximum performance was observed at an intermediate kernel width. Corresponding plots for all six subjects are provided in the Supplementary materials Fig. 2.

Fig. 5 demonstrates the impact of image denoising for all subjects in the data set. For both reconstruction methods (Fig. 5(A) based on Mika's method and Fig. 5(B) based on Kwok and Tsang's method) we observe an increase in model performance across all subjects, i.e. decrease in the minimum distance from the pr-curve to the point (1,1). In the prediction plots (column 2) the image denoising tends to induce both slightly increases and decreases in prediction accuracy for most subjects, whereas a more dramatic decrease is observed in subject 4. The reproducibility plots (column 3) show a prominent increase in reproducibilities in most subjects. We also observe an increase in the reproducibility of the sensitivity map for all subjects (column 4). Note that model selection was based on the minimum distance from the pr-curve to the point (1,1). Hence, the decrease in prediction accuracy for subject 4 (Fig. 5 column two), is fully compensated by the increased reproducibility (Fig. 5 column three) leading to a general decrease in distance (Fig. 5 column one). As a statistical test of the impact of image denoising we used a nonparametric Wilcoxon Signed Rank Test. For all measures except prediction accuracy we could reject the null-hypothesis, that the median difference between pairs of preprocessing methods was zero, at significance level 0.05. Hence, denoising leads to changes in the minimum distance from the pr-curve to the point (1,1), reproducibility of the FDA basis, and reproducibility of the sensitivity map, while prediction accuracy was not significantly affected.

Illustration II — manifold navigation

Fig. 6 provides the results of the manifold navigation illustration, where we visualize transitions from one class to another. Fig. 6(A) shows pre-images obtained along the line connecting the left- and right centroids in feature space for the finger tapping data set. In general, we observe 'activation' in contralateral regions of SMC and SMA (slice 49) and S2 (slice 13) and ipsilateral regions in cerebellum (the negative 'activation' is due to the centering of the data). In Fig. 6(C) we show pre-images projected onto the subspace defined by the first two basis images obtained by linear PCA, where the crosses corresponds to pre-images corresponding to points along the centroid connecting line in feature space. In general, the pre-image projections form a line connecting the outermost points. This linear behavior is in correspondence with the relatively linear kernel used as seen in Fig. 2. Fig. 6(B) is based on the object recognition data set. The top row shows pre-images obtained along the line connecting the bottle- and scissors centroids in feature space, the middle row shows pre-images obtained along the line connecting the scissors- and shoe centroids, and the bottom row shows pre-images along the line connecting the bottle- and shoe centroids. Note that some of the pre-images are identical. E.g. the pre-images corresponding to μ_1 in the top row and the middle row are identical, whereas the pre-images corresponding to μ_2 in the middle row and the bottom row are identical. Finally, The top row pre-image at μ_2 is identical to the pre-image in the bottom row corresponding to μ_1. These images are identical, since the pre-images were defined by moving along lines defined by centroids. Fig. 6(D) shows the pre-images projected onto the subspace obtained as in Fig. 6(C). We observe that the pre-images projections form a curve in the linear PCA subspace reflecting the use of a nonlinear kernel as seen in Fig. 4. Also note that while the pre-images are constructed from equidistant sampled points along the line defined in feature space the pre-image projections do not appear with equivalent distances in input space. In the Supplementary material we provide trajectories for all six subjects. Additionally, in Supplementary materials Figs. 3–4 we provide plots of distances between feature space points, from which we construct pre-images, and training points' projections in feature space. Generally such distances increases when moving away from the class centers.

Discussion

In order to demonstrate the use of kernel PCA and the subsequent pre-image reconstruction within the field of neuroimaging we have analyzed two data sets. The first is a simple finger tapping experiment with a relatively high signal to noise ratio, where the underlying networks and hence the spatial representation likely to best support discriminative information is relatively well understood. The other data set stems from an experiment on object representation, where the spatial pattern differences supporting discriminative information are likely to be more subtle. The finger tapping data were used to evaluate whether kernel PCA and pre-image reconstruction allow for identification of reliable models, and the object recognition data set to evaluate if our conclusions also holds in a data set with a more difficult classification task and with different preprocessing, e.g., no spatial smoothing.

In the finger tapping experiment we found that the kernel PCA subspace dimensionality had an impact on model performance, supporting a low dimensional data representation, while the preference towards relatively large widths of the Gaussian kernel suggested fairly linear kernel PCA subspaces. The preference towards linear models may reflect potential large variations in brain response patterns/localization across subjects, and the relatively simple classification

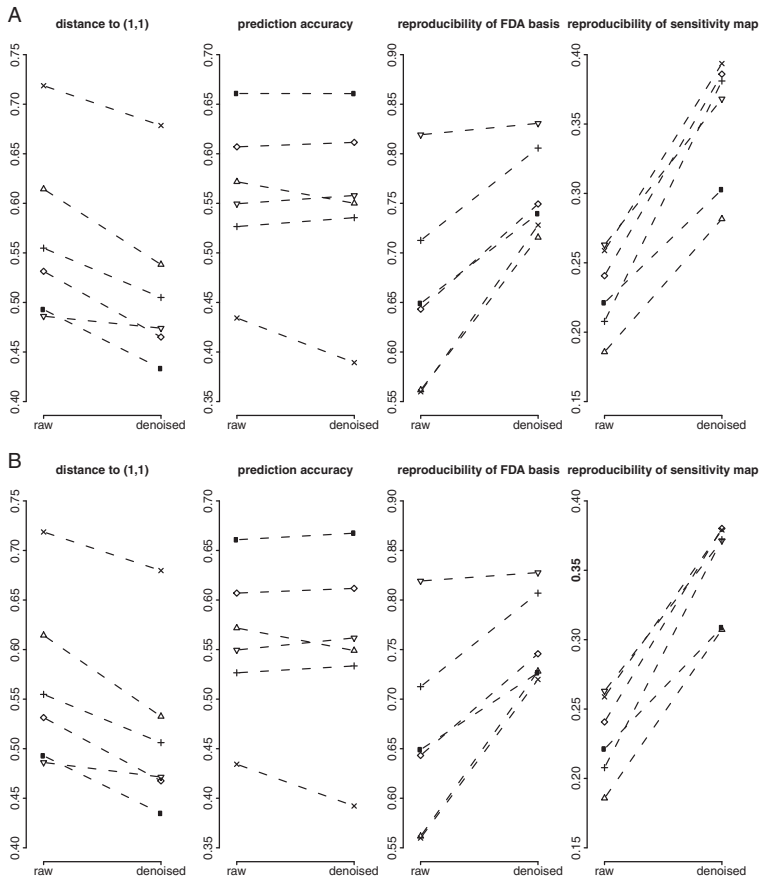


Fig. 5. Effect of denoising in the object recognition data set — changes in model performance at the subject level. Panel (A) is based on Mika's image reconstruction method and panel (B) is based on Kwok and Tsang's method. Comparisons are based on pr-maximizing models with denoising parameters selected across the denoising parameter grid (on a training set) (see Fig. 4), and pr-maximizing models build on the raw data. The first column shows model performance measured as the minimum distance from the pr-maximizing point on the pr-curve to (1,1) (see the Resampling and model evaluation section). The second column shows prediction accuracy, the third column shows pattern reproducibility — both measured at the pr-maximizing point, and the fourth column shows the reproducibility of the corresponding sensitivity map. The symbols {◦,△,+,×,◇,} correspond to subjects 1–6.

task. However, the results are not trivial, since there is no guarantee of reliable model identification following the image denoising. In the finger tapping data set we observed an increase in model performance in terms of increased reproducibility of classification models build on denoised data as measured within the NPAIRS resampling framework. The classification models build on denoised data identified cerebellar regions, subcortical regions, secondary supplementary motor area, sensorimotor cortex and supplementary motor areas as the underlying spatial pattern supporting discriminative information. Additionally, the $\overline{rSPI(Z)}$ Z-scores were increased for models built on the denoised data in comparison to models built on the raw data. We interpret the increases in Z-scores as indication of a more stable signal identification, since the Z-scores reflects voxel-wise signal to noise ratio (SNR) of the models' visualizations as estimated across independent splits of the data. Note that, in general, neither more identified voxels nor higher Z-scores directly imply better model performance, since we do not know the *true* relevant underlying brain network. However, note that our identification of the spatial layout of the brain pattern supporting discriminative information to the classifier is in agreement with previous research (Moritz et al., 2000a,b; Kustra and Strother, 2001; Riecker et al., 2003; Eickhoff et al., 2005; Witt et al., 2008).

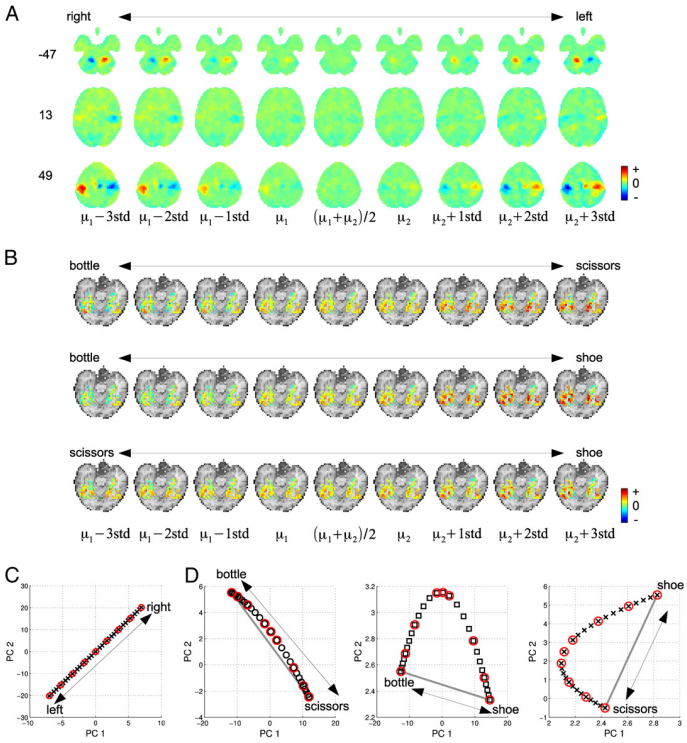


Fig. 6. Exploring the data structure by manifold navigation. The brain slices show pre-image reconstructions of feature space points along the line connecting the two class means μ_i (in feature space). The slices in panel (A) show interpolations/extrapolations between the left- and right conditions in the finger tapping data set. Panel (B) shows bottle↔scissors, bottle↔shoe, and scissors↔shoe transitions in the object recognition data set (subject 1). Numbers left to the slices denote z coordinates in MNI space in panel (A) while panel (B) is based on slice 27 in the volumes of subject 1. Panel (C) and (D) show projections of pre-images onto the subspace defined by the first two (linear) PCA basis images. The markers denote pre-images of feature space points along the line connecting the two class means. The markers highlighted with red circles correspond to pre-images at feature space locations $\{\mu_1-3\text{std},\mu_1-2\text{std},\ldots,\mu_1,(\mu_1+\mu_2)/2,\mu_2,\ldots,\mu_2+2\text{std},\mu_2+3\text{std}\}$. μ_i are projections of class means onto the line connecting the two class means, and std denote the standard deviation of the distribution of the projected points. The gray lines connect the outermost pre-images in the two classes.

In the manifold navigation illustration we found, that the trajectory of the pre-images' projections onto the line connecting the (RIGHT) and (LEFT) condition was quite linear. Hence, the pre-images are close to showing multiples of the class means. This illustrates a scenario that is similar to analyses that can be conducted with existing methodologies (Sato et al., 2008).

In the object recognition data set we observed, for all subjects, an increase in model performance as measured by the distance between the pr-maximizing point on the pr-curve and the point (1,1). In general this performance increase was based on relatively large increases in the reproducibilities of the spatial brain patterns extracted from the classification models. We found support of a relatively low dimensionality of the kernel PCA subspace, and best performance was observed at intermediate widths of the Gaussian kernel supporting a potentially nonlinear subspace as seen in Fig. 4. In the manifold navigation illustration we found pre-image trajectories that were quite nonlinear. When moving away from the class centers the distances between feature space points, from which we construct pre-images, and training points' projections in feature space increase (Supplementary materials Figs. 5–6). This effect has also been observed in other applications of kernel PCA and pre-image analysis (Kim et al., 2005). These authors suggested that kernel PCA is capable in learning signal structure in regions where training points are sampled dense. Hence, it should be noted that the outer most points along the trajectories in Fig. 6(D) are based on sparse regions of the feature space.

In order to simplify the manifold navigation illustration we performed selection of the width of the Gaussian kernel and kernel PCA subspace dimensionality based on the results in Fig. 4. Hence, the denoising parameters were identified based on the best performing classification model, as identified by considering the minimum distance from the pr-curve to (1,1) within the NPAIRS framework. The underlying signal structure may be well characterized in a space of lower dimensionality, than the one we here considered. Figs. 6(B,D) illustrates a scenario, where the proposed methodology allows for extraction of information, that cannot be identified with existing visualization procedures.

It is important to emphasize, that we do not attempt to make any neuroscientific claims about object representation in the human ventral temporal cortex based on the results that we here provide. Such an analysis would require e.g. assessment of the stability of the identified signal structure across multiple runs or subjects and is beyond the scope of the present article. Here we restrict our analysis to i) show (in Fig. 4) that the use of nonlinear modeling may lead to increases in model performances, and ii) that pre-image estimation allows for exploration of a nonlinear embedding manifold (Figs. 6(B,D)).

In the finger tapping data set we observed a slightly better performance by Mika's reconstruction method in comparison to Kwok and Tsang's method. In the object recognition data set we observed a quite similar performance when using Kwok and Tsang's or Mika's method. Importantly, we observed that a relatively large number of nearest neighbors (~500 out of 864) was required in Kwok and Tsang's method in the object recognition in order to achieve good model performance in terms of prediction accuracy and pattern reproducibility. Hence, it may be important to carefully select the number of nearest neighbors (along with the width of the Gaussian kernel and the kernel PCA subspace dimensionality) and not use a 'default' number of neighbors.

Methodological considerations

In this study we evaluated the impact of kernel PCA and the subsequent pre-image estimation as a preprocessing step on real fMRI data rather than in simulated data. Model evaluation was performed within the NPAIRS framework that has been introduced as an alternative to using receiver operating characteristic (ROC) curves based on simulated data. By using the finger tapping data set, our aim was to evaluate the denoising technique in a data set that reflects real neuroimaging settings, while maintaining a relatively easy classification task and a relatively well understood brain pattern underlying the brain states as defined by the experimental paradigm.

When building models within the NPAIRS resampling framework we based model selection on minimization of the distance from the pr-curve to the point (1,1) in the pr-plot. Hence, we focus not only on classification models that are highly predictive, but we also prefer models with high consistency in the spatial brain patterns extracted from the models. There are several issues that need to be addressed in this context. First, an increase in pattern reproducibility may not only be linked to an improved identification of the brain patterns supporting discriminative information. Increased reproducibility may also reflect noise structure/artifacts in the data that are reproducible across independent splits of the data (Strother et al., 2004; Yourganov et al., 2011). In the finger tapping data set we observed, that image denoising led to a general increase in the number of super-threshold voxels in the FDR thresholded maps and increases in $\overline{rSPI(Z)}$ Z-scores in brain regions correspond to regions identified in previous research. Second, by minimizing the Euclidean distance to the point (1,1) in the pr-plot we assign equal weight to the importance of prediction accuracy and pattern reproducibility. It has previously been demonstrated in simulated and real fMRI data, that maximizing pattern reproducibility while maintaining a high prediction accuracy may serve as a reasonable means for optimum signal detection, see e.g. Yourganov et al. (2011) and references therein.

In the analysis of the object recognition data set the model performance in terms of prediction accuracy is somewhat low compared to what has been reported in previous studies of the data set, e.g., by Hanson et al. (2004), that used a leave one out (LOO) cross validation resampling scheme. Here we apply split-half resampling in order to perform model evaluation within the NPAIRS resampling scheme, so that the reproducibility of the spatial patterns extracted from models trained on independent splits of the data can be evaluated. In the validation of our classification setup we were able to obtain LOO accuracies at the same level as has been reported in Hanson et al. (2004). Hence, by training the classifier on a larger training set we obtain a higher classification accuracy estimate. For a discussion of this issue in terms of learning curves we refer to Kjems et al. (2002).

In the Supplementary material we also provide plots similar to Fig. 6(D) for the remaining subjects in the object recognition data set. Note that the 'degree' of nonlinearity in the trajectories differs across subjects. We suggest the following explanation to this issue. i) The nonlinear manifolds are identified on individual subject's level. Hence, the manifold structure is not the same across subjects. The manifold structure will in general depend on the data observations present in the sample of each subject. ii) We project the trajectories onto the first two 'linear' PCs. However, the directions defined by the PCs are not the same across subject, and the PCs cannot be directly compared across subjects. iii) Even if the trajectories look linear, they may be 'more' nonlinear in other two-dimensional projections.

In the present work we have provided proof-of-concept illustrations of the use of kernel PCA and pre-image estimation as means for image denoising. A comprehensive investigation of how the kernel PCA/pre-image estimation procedure interacts with the other components of the preprocessing pipeline (LaConte et al., 2003) and compares to other denoising methods e.g. ICA or RETROICOR (Glover et al., 2000) is beyond the scope of the present study and is a topic for future research.

Future applications

In the present paper we have presented applications of pre-image estimation within the field of neuroimaging. Linear PCA has previously been demonstrated as a useful feature extractor (Bullmore et al., 1996; Hansen et al., 1999; Thomas et al., 2002). By using nonlinear features as in kernel PCA one may extract more information than available with linear features as discussed by Mika et al. (1999). Kernel PCA may be implemented in the data processing/analysis pipeline in several ways. i) López et al. (2009) used kernel PCA for feature extraction followed by building classifiers on the kernel PCA feature representation. This approach allows the investigator to estimate the generalization performance of the algorithm, but does not directly reveal where in the brain the discriminative information resides. ii) Another approach is to identify a low dimensional representation of the data with kernel PCA followed by pre-image estimation and building classifiers on the data reconstructed in input/voxel space. In the present study the subspace was identified by a series of principal components sorted according to decreasing variance. An alternative strategy is to construct a relevant signal subspace by identification of the components that are relevant to the discriminative task, i.e., instead of retaining the q components with the largest variance we may retain q components according to their relative importance in solving the classification task. iii) Furthermore, building a nonlinear classifier directly on the data represented in voxel space, voxels containing relevant information to the classifier could be identified by the sensitivity map as investigated in Rasmussen et al. (2011). While the sensitivity map provides a *global* summary map, the pre-image analysis could allow for exploration of *localized* structures in feature space such as visualization of class centroids. Such an analysis would be similar to the hyperplane navigation method proposed by Sato et al. (2008) and the feature space navigation method that we outline in the present study. Exploration of a low dimensional embedding allows for interpolation/prediction of 'unknown'/intermediate brain states (Raizada and Kriegeskorte, 2010) and exploration of the continuum between defined brain states. In this context pre-image estimation provides a voxel space representation of such intermediate points. iv) Finally, kernel PCA denoising and pre-image reconstruction may be implemented in on-line analysis settings, where pre-image reconstruction allows monitoring of the instant brain state/activation pattern.

Conclusion

In the present proof of concept study we have demonstrated the use of unsupervised nonlinear denoising by kernel PCA and subsequent pre-image reconstruction within the field on neuroimaging. As a denoising step in the preprocessing pipeline, kernel PCA and pre-image estimation lead to improved performance as measured by prediction performance and reproducibility of spatial maps extracted from classification models within the NPAIRS resampling framework. As an objective for model selection we focused on jointly maximizing prediction accuracy and pattern reproducibility. We found, that the image denoising lead to relatively large gains in the reproducibility of the spatial maps while maintaining the predictive performance. Additionally, we have shown that pre-image estimation allows for exploration of a nonlinear embedding manifold by generation of brain maps in the continuum between defined experimental brain states/classes.

Acknowledgments

We thank the authors of Haxby et al. (2001) for making the data set on face and object representation in the human ventral temporal cortex available under the terms of the Creative Commons Attribution-Share Alike 3.0 license. We also thank the PyMVPA (http://www.pymvpa.org) developers for making the data set publicly available. We thank our anonymous reviewers for their many constructive comments that allowed us to improve the readability and quality of the paper. We thank Torben E. Lund, The Danish National Research Foundation's Center for Functionally Integrative Neuroscience, Aarhus University Hospital, Denmark, and Stephen C. Strother, The Rotman Research Institute Baycrest, Toronto, Canada, for several comments on our work. The Simon Spies Foundation is acknowledged for donation of the Siemens Trio scanner. Kristoffer H. Madsen was supported by the Danish Medical Research Council (grant no. 09-072163) and the Lundbeck Foundation (grant no. R48-A4846).

Appendix A. Matlab code available

Matlab code for demonstration of kernel PCA and pre-image estimation is available from the website http://code.google.com/p/kpca-fmri/. Alternatively, the code can be requested by contacting the first author by email. The code demonstrates denoising and the manifold navigation procedure on the object recognition data set.

Appendix B

For very small values of the width parameter (very nonlinear kernels), the off-diagonal elements of the Gaussian kernel matrix, K_G, approaches 0, while all diagonal elements approach 1. This follows from the definition, $k_G(x_i,x_j) = exp(-\frac{1}{\sigma}||x_i-x_j||^2)$, and letting $\sigma \ll max||x_i-x_j||^2$. Thus, the kernel matrix approaches the identity matrix. Since the kernel function is defined as the inner-product of the observations, $k_G(x_i,x_j) = \varphi(x_i)^T\varphi(x_j)$, and an inner-product of zero implies orthogonality, all $\varphi(x_i)$ will be (almost) orthogonal to each other in this case. Hence, PCA will be meaningless.

In the other extreme, as σ approached ∞ and the smoothing is large, it seems intuitive that kernel PCA approaches linear PCA. The claim is that for $\sigma \gg max||x_i-x_j||^2$ the centered kernel matrix for a Gaussian kernel will be similar to that of a linear kernel of the form $k_L(x_i,x_j) = x_i^Tx_j$. It is easily seen that in the linear case, the kernel matrix is similar to the covariance matrix in input space, and hence the eigendecomposition of $K_L$will resemble that for linear PCA.

In order to investigate the kernel matrix for the Gaussian kernel when $\sigma \to \infty$ multivariate Taylor expansion of the Gaussian kernel is performed. Due to the assumption $\sigma \gg max||x_i-x_j||^2$, terms smaller than $O(\frac{1}{\sigma})$ can be ignored, leading to the following second order expansion:

$$k_G\left(x_i,x_j\right) = exp\left(-\frac{1}{\sigma}||x_i-x_j||^2\right) \approx 1-\frac{||x_i-x_j||^2}{\sigma}$$

Now, the centered kernel matrix is given by

$$\tilde{k}\left(x_i,x_j\right) = k\left(x_i,x_j\right) - \frac{1}{N}1_N^Tk_x - \frac{1}{N}1_N^Tk_{x'} + \frac{1}{N^2}1_N^TK1_N$$

where $k_x = [k(x,\boldsymbol{x}_1),\ldots,k(x,x_N)]^T$ is a column vector of kernel functions of x and the training set, and 1_N is a N-dimensional column vector of ones. Inserting the Taylor expansion into this expression gives the approximate centered kernel function of a very smooth Gaussian kernel:

$$\tilde{k}_G\left(x_i,x_j\right) \approx \frac{2}{\sigma}\left(x_i^Tx_j - \frac{1}{N}\sum_{n=1}^{N}x_i^Tx_n - \frac{1}{N}\sum_{n=1}^{N}x_n^Tx_j + \frac{1}{N^2}\sum_{n,m=1}^{N}x_n^Tx_m\right) \quad (B.1)$$

The expression within the brackets can easily be recognized as the centering of the linear kernel function. Hence

$$\tilde{K}_G \approx \frac{2}{\sigma}\tilde{K}_L = \frac{2}{\sigma}\tilde{X}^T\tilde{X}$$

Since $\Sigma = \frac{1}{N}\tilde{X}^T\tilde{X}$ it follows that kernel PCA will be identical to linear PCA up to a scaling of $2N/\sigma$ when $\sigma \to \infty$.

Appendix C

When using explicit centering, the projection onto the q-dimensional principal subspace can be expressed as $P_q\varphi(x) = \sum_{n=1}^{N}\tilde{\gamma}_n(\varphi(x_n)-\bar{\varphi}) + \bar{\varphi}$, where the mean $\bar{\varphi} = \sum_{n=1}^{N}\varphi(x_n)/N$. By expansion, the cost function can be rewritten as:

$$\begin{aligned}
&||\varphi(z)-P_q\varphi(x)||^2 \\
&= k(z,z) - 2\sum_{n=1}^{N}\left(\tilde{\gamma}_n k(z,x_n) - \frac{1}{N}\tilde{\gamma}_n\sum_{m=1}^{N}k(z,x_m) + \frac{1}{N}k(z,x_n)\right) + \Omega \\
&= k(z,z) - 2\sum_{n=1}^{N}\gamma_n k(z,x_n) + \Omega
\end{aligned}$$

where all the z-independent terms (stemming from $||P_q\phi(x)||^2$) have been collected in Ω. The last equality follows from letting $\gamma_n = \tilde{\gamma}_n + \frac{1}{N}\left(1-\sum_{j=1}^{N}\tilde{\gamma}_j\right)$, where $\tilde{\gamma}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$ as defined in Eq. (1).

Appendix D

In Algorithm 1 we outline the NPAIRS resampling procedure for estimating an average reproducible brain map $\overline{rSPI(Z)}$, prediction accuracy p, and reproducibility r . Let $\mathcal{D}$ denote the data set, split$(\cdot)$ denote the operation of splitting a set into two disjoint sets, $f(\cdot)$ denote the model building process, $g(\cdot)$ denote the process of evaluating test set accuracy, $h(\cdot)$ denote the process of extracting a visualization (brain map) from a model.

Algorithm 1. Calculate $\overline{rSPI(Z)}$, p, and r

for $i = 1 \to N_{splits}$ do
  $[\mathcal{S}_1,\mathcal{S}_2] \leftarrow$ split$(\mathcal{D})$
  $\mathcal{M}_1 \leftarrow f(\mathcal{S}_1)$
  $\mathcal{M}_2 \leftarrow f(\mathcal{S}_2)$
  $a_1 \leftarrow g(\mathcal{M}_1,\mathcal{S}_2)$
  $a_2 \leftarrow g(\mathcal{M}_2,\mathcal{S}_1)$

$p_i \leftarrow (a_1 + a_2)/2$
$w_1 \leftarrow h(\mathcal{M}_1)$
$w_2 \leftarrow h(\mathcal{M}_2)$
$r_i \leftarrow \mathrm{corr}(w_1, w_2)$
$w_1 \leftarrow w_1/\mathrm{std}(w_1)$
$w_2 \leftarrow w_2/\mathrm{std}(w_2)$
$s \leftarrow (w_1 + w_2)/\sqrt{2}$
$n \leftarrow (w_1 - w_2)/\sqrt{2}$
$rSPI(Z)_i \leftarrow s/\mathrm{std}(n)$
end for
$\overline{rSPI(Z)} \leftarrow \sum_i rSPI(Z)_i / N_{splits}$
return $\overline{rSPI(Z)}, p, r$

Appendix E. Supplementary data

Supplementary data to this article can be found online at doi:10.1016/j.neuroimage.2012.01.096.

References

Abrahamsen, T.J., Hansen, L.K., 2009. Input space regularization stabilizes pre-images for kernel PCA de-noising. IEEE Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6.

Abrahamsen, T., Hansen, L., 2011. Regularized pre-image estimation for kernel PCA de-noising. J. Signal Process. Syst. 65, 403–412.

Arias, P., Randall, G., Sapiro, G., 2007. Connecting the out-of-sample and pre-image problems in kernel methods. 18-23 jun IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. Royal Statistical Society. Series B (Methodological) 57 (1), 289–300.

Bullmore, E.T., Rabe-Hesketh, S., Morris, R.G., Williams, S.C.R., Gregory, L., Gray, J.A., Brammer, M.J., 1996. Functional magnetic resonance image analysis of a large-scale neurocognitive network. NeuroImage 4 (1), 16–33.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2, 121–167.

Chen, X., Pereira, F., Lee, W., Strother, S., Mitchell, T., 2006. Exploring predictive and reproducible modeling with the single-subject FIAC dataset. Hum. Brain Mapp. 27 (5), 452–461.

Dambreville, S., Rathi, Y., Tannenbaum, A., 2006. Statistical shape analysis using kernel PCA. IS&T/SPIE Symposium on Electrical Imaging, Vol. 6064(B).

Eickhoff, S.B., Amunts, K., Mohlberg, H., Zilles, K., 2005. The human parietal operculum. II. Stereotaxic maps and correlation with functional imaging results. Cereb. Cortex 16 (2), 268–279.

Friston, K.J., Holmes, A.P., Worsley, K.J., Poline, J.P., Frith, C.D., Frackowiak, R.S.J., 1994. Statistical parametric maps in functional imaging: a general linear approach. Hum. Brain Mapp. 2 (4), 189–210.

Glover, G.H., Li, T.Q., Ress, D., 2000. Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. Magn. Reson. Med. 44 (1), 162–167.

Guo, Y., 2010. A weighted cluster kernel PCA prediction model for multi-subject brain imaging data. Stat. Interface 3 (1), 103–111.

Hansen, L.K., Larsen, J., Nielsen, F.A., Strother, S.C., Rostrup, E., Savoy, R., Lange, N., Sidtis, J., Svarer, C., Paulson, O.B., 1999. Generalizable patterns in neuroimaging: how many principal components? NeuroImage 9 (5), 534–544 May.

Hanson, S.J., Matsuka, T., Haxby, J.V., 2004. Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a "face" area? NeuroImage 23 (1), 156–166.

Hastie, T., Buja, A., Tibshirani, R., 1995. Penalized discriminant analysis. Ann. Stat. 23, 73–102.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer.

Haxby, J.V., Gobbini, I.M., Furey, M.L., Ishai, A., Schouten, J.L., Pietrini, P., 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 293 (5539), 2425–2430.

Kim, K.I., Franz, M.O., Schölkopf, B., 2005. Iterative kernel principal component analysis for image modeling. IEEE Trans. Pattern Anal. Mach. Intell. 27 (9), 1351–1366.

Kjems, U., Hansen, L.K., Anderson, J., Frutiger, S., Muley, S., Sidtis, J., Rottenberg, D., Strother, S.C., 2002. The quantitative evaluation of functional neuroimaging experiments: mutual information learning curves. NeuroImage 15 (4), 772–786.

Kustra, R., Strother, S.C., 2001. Penalized discriminant analysis of [15-O]-water PET brain images with prediction error selection of smoothness and regularization hyperparameters. IEEE Trans. Med. Imaging 20 (5), 376–387.

Kwok, J.T.-Y., Tsang, I.W.-H., 2004. The pre-image problem in kernel methods. IEEE Trans. Neural Netw. 15 (6), 1517–1525.

LaConte, S., Anderson, J., Muley, S., Ashe, J., Frutiger, S., Rehm, K., Hansen, L.K., Yacoub, E., Hu, X., Rottenberg, D., Strother, S., 2003. The evaluation of preprocessing choices in single-subject BOLD fMRI using NPAIRS performance metrics. NeuroImage 18 (1), 10–27.

López, M., Ramírez, J., Górriz, J., Álvarez, I., Salas-Gonzalez, D., Segovia, F., Chaves, R., 2009. SVM-based CAD system for early detection of the Alzheimer's disease using kernel PCA and LDA. Neurosci. Lett. 464 (3), 233–238.

McKeown, M.J., Jung, T.-P., Makeig, S., Brown, G., Kindermann, S.S., Lee, T.-W., Sejnowski, T.J., 1998. Spatially independent activity patterns in functional MRI data during the Stroop color-naming task. Proc. Natl. Acad. Sci. 95, 803–810.

Mika, S., Schölkopf, B., Smola, A., Müller, K.R., Scholz, M., Rätsch, G., 1999. Kernel PCA and de-noising in feature spaces. Advances in Neural Information Processing Systems, 11. MIT Press, pp. 536–542.

Moritz, C.H., Haughton, V.M., Cordes, D., Quigley, M., Meyerand, M.E., 2000a. Whole-brain Functional MR Imaging activation from a finger-tapping task examined with independent component analysis. Am. J. Neuroradiol. 21 (9), 1629–1635.

Moritz, C.H., Meyerand, M.E., Cordes, D., Haughton, V.M., 2000b. Functional MR Imaging activation after finger tapping has a shorter duration in the basal ganglia than in the sensorimotor cortex. Am. J. Neuroradiol. 21 (7), 1228–1234.

Raizada, R.D.S., Kriegeskorte, N., 2010. Pattern-information fMRI: new questions which it opens up and challenges which face it. Int. J. Imaging Syst. Technol. 20 (1), 31–41.

Rasmussen, P.M., Hansen, L.K., Madsen, K.H., Chruchill, N.W., Strother, S.C., 2012. Model sparsity and brain pattern interpretation of classification models in neuroimaging. Pattern Recognit. 45 (6), 2085–2100.

Rasmussen, P.M., Madsen, K.H., Lund, T.E., Hansen, L.K., 2011. Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. NeuroImage 55 (3), 1120–1131.

Riecker, A., Wildgruber, D., Mathiak, K., Grodd, W., Ackermann, H., 2003. Parametric analysis of rate-dependent hemodynamic response functions of cortical and subcortical brain structures during auditorily cued finger tapping: a fMRI study. NeuroImage 18 (3), 731–739.

Sato, J.R., Thomaz, C.E., Cardoso, E.F., Fujita, A., Martin, M.D.G.M., Amaro Jr., E., 2008. Hyperplane navigation: a method to set individual scores in fMRI group datasets. NeuroImage 42 (4), 1473–1480.

Schlökopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L., 2000. New support vector algorithms. Neural Comput. 12 (5), 1207–1245.

Schölkopf, B., Smola, A., Müller, K.-R., 1998a. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10 (5), 1299–1319.

Schölkopf, B., Smola, A.J., Knirsch, P., Burges, C.J.C., 1998b. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. Mustererkennung 1998, 20. DAGM-Symposium. Springer-Verlag, pp. 125–132.

Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J., 1999. Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. 10 (5), 1000–1017.

Smith, S., Jenkinson, M., Woolrich, M., Beckmann, C., Behrens, T., Johansen-Berg, H., Bannister, P., De Luca, M., Drobnjak, I., Flitney, D., Niazy, R., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J., Matthews, P., 2004. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage 26 (S1), 208–219.

Song, X., Ji, T., Wyrwicz, A.M., 2008. Baseline drift and physiological noise removal in high field fmri data using kernel pca. ICASSP, pp. 441–444.

Strother, S.C., Conte, S.L., Hansen, L.K., Anderson, J., Zhang, J., Pulapura, S., Rottenberg, D., 2004. Optimizing the fMRI data-processing pipeline using prediction and reproducibility performance metrics: I. a preliminary group analysis. NeuroImage 23 (Supplement 1), S196–S207.

Strother, S., Oder, A., Spring, R., Grady, C., 2010. The NPAIRS computational statistics framework for data analysis in neuroimaging. In: Saporta, G. (Ed.), 19th International Conference on Computational Statistics. Physica-Verlag, Heidelberg, Paris, pp. 111–120.

Strother, S.C., Anderson, J., Hansen, L.K., Kjems, U., Kustra, R., Sidtis, J., Frutiger, S., Muley, S., Laconte, S., Rottenberg, D., 2002. The quantitative evaluation of functional neuroimaging experiments: the NPAIRS data analysis framework. NeuroImage 15 (4), 747–771 April.

Thirion, B., Faugeras, O., 2003. Dynamical components analysis of fMRI data through kernel PCA. NeuroImage 20 (1), 34–49.

Thomas, C.G., Harshman, R.A., Menon, R.S., 2002. Noise reduction in BOLD-based fMRI using component analysis. NeuroImage 17 (3), 1521–1537 Nov..

Tohka, J., Foerde, K., Aron, A.R., Tom, S.M., Toga, A.W., Poldrack, R.A., 2008. Automatic independent component labeling for artifact removal in fmri. NeuroImage 39 (3), 1227–1245.

Witt, S.T., Laird, A.R., Meyerand, M.E., 2008. Functional neuroimaging correlates of finger-tapping task variations: an ALE meta-analysis. NeuroImage 42 (1), 343–356 Aug..

Yourganov, G., Chen, X., Lukic, A.S., Grady, C.L., Small, S.L., Wernick, M.N., Strother, S.C., 2011. Dimensionality estimation for optimal detection of functional networks in BOLD fMRI data. NeuroImage 56 (2), 531–543.

Zhang, J., Anderson, J.R., Liang, L., Pulapura, S.K., Gatewood, L., Rottenberg, D.A., Strother, S.C., 2009a. Evaluation and optimization of fMRI single-subject processing pipelines with NPAIRS and second-level CVA. Magn. Reson. Imaging 27 (2), 264–278.

Zhang, Z., Dai, G., Jordan, M.I., 2009b. A flexible and efficient algorithm for regularized fisher discriminant analysis. ECML PKDD'09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases. Springer-Verlag, Berlin, Heidelberg, pp. 632–647.

Appendix D

Information-based Kernel PCA Denoising by Semi-supervised Manifold Learning

---

Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. Information-based Kernel PCA Denoising by Semi-supervised Manifold Learning. *Pattern Recognition Letters*, Under review, 2013.

Information-based Kernel PCA Denoising by Semi-supervised Manifold Learning

Toke Jansen Hansen, Trine Julie Abrahamsen, Lars Kai Hansen

DTU Compute, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Abstract

Kernel Principal Component Analysis (PCA) has proven a powerful tool for nonlinear feature extraction, and is often applied as a pre-processing step for classification algorithms. In denoising applications Kernel PCA provides the basis for dimensionality reduction, prior to the so-called *pre-image problem* where denoised feature space points are mapped back into input space. This problem is inherently ill-posed due to the non-bijective feature space mapping. We present a semi-supervised denoising scheme based on kernel PCA and the pre-image problem, where class labels on a subset of the data points are used to improve the denoising. Moreover, by warping the Reproducing Kernel Hilbert Space (RKHS) we also account for the intrinsic manifold structure yielding a Kernel PCA basis that also benefit from unlabeled data points. Our two main contributions are; 1) A generalization of Kernel PCA by incorporating a loss term, leading to an iterative algorithm for finding orthonormal components biased by the class labels, and 2) A fixed-point iteration for solving the pre-image problem based on a manifold warped RKHS. We prove viablity of the proposed methods on both synthetic data and images from The Amsterdam Library of Object Images (Geusebroek et al., 2005).

Keywords: Semi-supervised denoising, kernel PCA, pre-image problem

1. Introduction

In Principal Component Analysis (PCA) we seek an orthogonal basis that maximizes the explained variance of a data set. This basis can be found by

Email address: tjha@imm.dtu.dk, +45 45253888 (Toke Jansen Hansen)

computing eigenvectors of the centered covariance matrix, where the magnitude of an eigenvalue λ_i equals the amount of variance in the direction of the corresponding eigenvector $\boldsymbol{v}_i$, also denoted as the i^{th}*principal component*. In data compression, data is represented by a subset of the principal components having the largest eigenvalues, thereby ensuring that we retain as much variance as possible, whereas in denoising applications we deliberately drop directions with small variance (Mika et al., 1999).

When the data set contain nonlinear structures we cannot rely on linear PCA to provide a meaningful representation. Kernel PCA is the natural generalization of PCA, leveraging on the well known *kernel trick* to explain complicated nonlinear relations. We can think of the kernel PCA procedure as employing a function $\varphi : \mathcal{X} \mapsto \mathcal{H}$ that maps data from a $D_{\mathcal{X}}$-dimensional input space $\mathcal{X}$ to a $D_{\mathcal{H}}$-dimensional feature space $\mathcal{H}$ (possibly infinite dimensional), followed by performing linear PCA in $\mathcal{H}$. In practice we do never carry out the explicit mapping, but instead exploit the *kernel trick* stating that inner products in $\mathcal{H}$ can be computed in terms of kernel evaluations in $\mathcal{X}$, *i.e.*, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)$. Hence, all algorithms that can be formulated solely in terms of inner products are applicable for the kernel trick, where the function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ must fulfill Mercers condition, stating that $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ must be a positive definite matrix. A popular choice of kernel function is the Gaussian, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$, that has been successfully applied in both classification and denoising applications (Schölkopf et al., 1998).

For denoising purposes, we are interested in estimating the inverse mapping, φ^{-1}, known as the *pre-image problem*. For the Gaussian kernel the implicit mapping defined by φ is non-bijective leading to the inherently ill-posed pre-image problem. The fixed-point iteration described by Mika et al. (1999), provides an efficient scheme for determining the pre-images for Gaussian kernels, building upon standard gradient descent methods.

In this contribution we apply semi-supervised learning to construct a label informed kernel PCA basis. We achieve this, by extending the kernel PCA objective with a loss term and derive an efficient algorithm for computing an orthonormal basis biased towards a set of labeled training points. Furthermore, we derive a fixed-point iteration for finding an approximate pre-image for the kernel function introduced by Sindhwani et al. (2005). This Graph based kernel warps the corresponding RKHS to account for the manifold structure imposed by both labeled and unlabeled data points. The common goal for these two methods is to exploit labeled data to determine a more descriptive manifold representation. I.e., when using a fixed number of com-

ponents we claim to achieve "better" denoised reconstructions than standard kernel PCA.

1.1. Related work

There is a vast literature on both kernel methods and semi-supervised learning, hence, for a general overview we refer to, e.g., Chapelle et al. (2006). The pre-image problem was initially studied by Mika et al. (1999), who derived a fixed-point iteration for the Gaussian kernel. Bakir et al. (2004b) considered the pre-image problem for undirected graphs, and suggested a scheme for reconstructing graphs from the RKHS representation. Later studies considered regularization to make the pre-image problem more well behaved, see for instance Abrahamsen and Hansen (2011).

Walder et al. (2010) introduced the notion of semi-supervised kernel PCA by including a loss term, and derived solutions for objectives based on both squared and logistic losses. In particular, the squared loss can be interpreted as the Spectral Graph Transducer (SGT) by Joachims (2003), when the RKHS is defined by a graph based regularizer. In terms of the objective both Walder et al. (2010) and Joachims (2003) consider variations of a constrained eigenvalue problem and rely on a neat result by Gander et al. (1989) for a unique closed-form solution.

Another way of incorporating label information was introduced by Sindhwani et al. (2005) through the idea of warping the RKHS to account for the manifold structure imposed by both labeled and unlabeled data points, and derived the kernel

$$\tilde{k}(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{k}_x^\top (\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})^{-1} \boldsymbol{L} \boldsymbol{k}_y \tag{1}$$

where $\boldsymbol{k_y} = [k(\boldsymbol{y}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{y}, \boldsymbol{x}_N)]^\top$ and $\boldsymbol{L}$ is the combinatorial graph Laplacian, defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{K}$, where $\boldsymbol{D}$ is a diagonal matrix with $D_{ii} = \sum_{i=1}^{N} K_{ij}$. In the remainder of this paper, we will denote the above kernel function as the *Graph kernel*.

Our work can be considered extensions of Walder et al. (2010) and Joachims (2003), in that we generalize the objective with an orthogonality constraint to enable the construction of more than one orthogonal basis vectors. Furthermore, we derive a fixed-point iteration for the pre-image problem based on the Graph kernel by Sindhwani et al. (2005), that directly relates to the SGT. However, we emphasize that generalizing the semi-supervised kernel PCA objective to allow for an arbitrary number orthogonal components is

relevant, along the same line as extracting more than a single kernel PCA component when higher dimensional representations are needed to describe the signal manifold.

2. Methods

The remainder of this section will be outlined as follows. In Section 2.1 we extend the usual kernel PCA objective with a squared loss term, similar to the work of Walder et al. (2010), and develop a scheme for finding a semi-supervised kernel PCA basis of arbitrary dimensionality. In Section 2.2 we leverage on the ideas of Sindhwani et al. (2005) and apply them in the context of the pre-image problem, by deriving a fixed-point iteration for Graph kernel.

2.1. Semi-supervised kernel PCA

In semi-supervised kernel PCA we incorporate knowledge of the class labels on a subset of the data points. In this section we generalize the result of Walder et al. (2010) to account for multiple orthonormal components, thereby allowing us to compute a kernel PCA basis where the n^{th} direction is biased towards training labels with the constraint of being perpendicular to the previous $n-1$ components.

Figure 1 shows the original kernel PCA objective together with our modification that incorporates a least squares loss term in the form of an additional constraint. Note that we are explicit about the kernel PCA components being perpendicular in both the original objective and our modification, since in the latter case this constraint must be handled by an explicit projection onto the null space of previous components.

For the original kernel PCA objective we can apply the representer theorem $f^\star(\cdot) = \sum_{i=1}^N \alpha_i^\star k(\boldsymbol{x}_i, \cdot)$ and form the derivative with respect to $\boldsymbol{\alpha}$ of the Lagrangian, leading to the following generalized eigenvalue problem

$$\boldsymbol{K}\boldsymbol{\alpha} = \lambda(\boldsymbol{K}^\top\boldsymbol{K} - \boldsymbol{K}^\top\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{E}_N$ is a matrix of size N with entries $\frac{1}{N}$.

To solve the extended semi-supervised objective efficiently we rewrite it in a similar manner as in Walder et al. (2010), where we minimize the norm

Kernel PCA	**Semi-supervised kernel PCA**
$\max_{f_n \in \mathcal{H}} \quad \sum_{i=1}^{N} \left(f_n(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} f_n(\boldsymbol{x}_j) \right)^2$	$\max_{f_n \in \mathcal{H}} \quad \sum_{i=1}^{N} \left(f_n(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} f_n(\boldsymbol{x}_j) \right)^2$
s.t $\quad \|f_n\|_{\mathcal{H}}^2 = 1$	s.t $\quad \|f_n\|_{\mathcal{H}}^2 = 1$
$\sum_{i=1}^{n-1} \langle f_n, f_i \rangle_{\mathcal{H}}^2 = 0$	$\sum_{i=1}^{n-1} \langle f_n, f_i \rangle_{\mathcal{H}}^2 = 0$
	$\sum_{i \in \mathcal{L}} (f_n(\boldsymbol{x}_i) - y_i)^2 \leq \omega$

Figure 1: Left: The usual kernel PCA objective. Right: Our modified kernel PCA objective incorporating a least squares loss term. $\mathcal{L}$ is the set of labeled training data and ω determines the allowed derivation from the true labels.

together with the squared loss term while keeping the variance fixed.

$$\underset{f_n \in \mathcal{H}}{\text{minimize}} \quad \|f_n\|_{\mathcal{H}}^2 + c \sum_{i \in \mathcal{L}} (f_n(\boldsymbol{x}_i) - y_i)^2 \tag{3}$$

$$\text{s.t} \quad \sum_{i=1}^{N} \left(f_n(\boldsymbol{x}_i) - \frac{1}{N} \sum_{j=1}^{N} f_n(\boldsymbol{x}_j) \right)^2 = s^2 \tag{4}$$

$$\sum_{i=1}^{n-1} \langle f_n, f_i \rangle_{\mathcal{H}}^2 = 0 \tag{5}$$

The main reason for the above formulation is that the linear part of the squared loss term makes the relationship between s and $f^\star$ non-trivial, but in this constellation we can control the relative importance of the respective terms via the parameters c and s^2. Applying the representer theorem yields

$$\underset{\boldsymbol{\alpha}_n \in \mathbb{R}^N}{\text{minimize}} \quad \boldsymbol{\alpha}_n^\top \boldsymbol{K} \boldsymbol{\alpha}_n + c \|\boldsymbol{K}_{\mathcal{L}} \boldsymbol{\alpha}_n - \boldsymbol{t}\|^2 \tag{6}$$

$$\text{s.t} \quad \boldsymbol{\alpha}_n^\top (\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K}) \boldsymbol{\alpha}_n = s^2 \tag{7}$$

$$\sum_{i=1}^{n-1} (\boldsymbol{\alpha}_n^\top \boldsymbol{K} \boldsymbol{\alpha}_i)^2 = 0 \tag{8}$$

where $\boldsymbol{t} \in \mathbb{R}^{|\mathcal{L}|}$ is a sub-vector of $\boldsymbol{y}$ that only takes indices $\mathcal{L}$, and likewise does $\boldsymbol{K}_{\mathcal{L}}$ denote the sub-matrix of $\boldsymbol{K}$ by taking rows $\mathcal{L}$. To account for

the orthogonality constraint in Equation (8) we apply a projection operator on $\boldsymbol{\alpha}_n$, forcing the solution to be in the null space of previous solutions, see for instance Golub (1973). Let $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{n-1}]$ be the previous components, then the $\boldsymbol{S} = \text{Null}(\boldsymbol{K}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{K})$ is an orthonormal basis of size $N \times (N - n + 1)$ for the null space of $\boldsymbol{K}\boldsymbol{A}$ obtained from a singular value decomposition (SVD). Hence, by projectin $\boldsymbol{S}\boldsymbol{\alpha}_n$, the Lagrangian of the semi-supervised kernel PCA problem in Equation (6)-(8) can be formulated as a $(N - n + 1)$-dimensional problem

$$\begin{aligned} L = &\boldsymbol{\alpha}_n^\top \boldsymbol{S}^\top \boldsymbol{K}\boldsymbol{S}\boldsymbol{\alpha}_n + c||\boldsymbol{K}_\mathcal{L}\boldsymbol{S}\boldsymbol{\alpha}_n - \boldsymbol{t}||^2 \\ &+ \lambda(\boldsymbol{\alpha}_n^\top \boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}\boldsymbol{\alpha}_n - s^2) \end{aligned} \tag{9}$$

Setting the partial derivatives to zero gives

$$\begin{aligned} \frac{\delta L}{\delta \boldsymbol{\alpha}_n} &= 2\boldsymbol{S}^\top \boldsymbol{K}\boldsymbol{S}\boldsymbol{\alpha}_n + 2c\boldsymbol{S}^\top \boldsymbol{K}_\mathcal{L}\boldsymbol{K}_\mathcal{L}\boldsymbol{S}\boldsymbol{\alpha}_n \\ &\quad - 2c\boldsymbol{S}^\top \boldsymbol{K}_\mathcal{L}\boldsymbol{t} + 2\lambda\boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}\boldsymbol{\alpha}_n \\ &= \boldsymbol{0} \end{aligned} \tag{10}$$

$$\frac{\delta L}{\delta \lambda} = \boldsymbol{\alpha}_n^\top \boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}\boldsymbol{\alpha}_n - s^2 = 0 \tag{11}$$

Leading to the following system of coupled equations

$$\begin{aligned} &\boldsymbol{S}^\top(\boldsymbol{K} + c\boldsymbol{K}_\mathcal{L}\boldsymbol{K}_\mathcal{L})\boldsymbol{S}\boldsymbol{\alpha}_n = \\ &-\lambda\boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}\boldsymbol{\alpha}_n + c\boldsymbol{S}^\top \boldsymbol{K}_\mathcal{L}\boldsymbol{t} \end{aligned} \tag{12}$$

$$\boldsymbol{\alpha}_n^\top \boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}\boldsymbol{\alpha}_n = s^2 \tag{13}$$

Substituting $\boldsymbol{C} = \boldsymbol{S}^\top(\boldsymbol{K} + c\boldsymbol{K}_\mathcal{L}\boldsymbol{K}_\mathcal{L})\boldsymbol{S}$, $\boldsymbol{b} = c\boldsymbol{S}^\top \boldsymbol{K}_\mathcal{L}\boldsymbol{t}$, and $\boldsymbol{P} = \boldsymbol{S}^\top(\boldsymbol{K}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N\boldsymbol{K})\boldsymbol{S}$, these simplify to

$$\boldsymbol{C}\boldsymbol{\alpha}_n = -\lambda\boldsymbol{P}\boldsymbol{\alpha}_n + \boldsymbol{b} \tag{14}$$

$$\boldsymbol{\alpha}_n^\top \boldsymbol{P}\boldsymbol{\alpha}_n = s^2 \tag{15}$$

The first equation leads to

$$\boldsymbol{\alpha}_n = (\boldsymbol{C} + \lambda\boldsymbol{P})^{-1}\boldsymbol{b} \tag{16}$$

To saturate the second equation we can make use of the ideas in Gander et al. (1989), stating that λ should equal the smallest eigenvalue of the following generalized eigenvalue problem

$$\begin{pmatrix} \boldsymbol{C} & -\boldsymbol{P} \\ -\frac{1}{s^2}\boldsymbol{b}\boldsymbol{b}^\top & \boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\eta} \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\eta} \end{pmatrix} \tag{17}$$

Typically this system is badly conditioned, so in practical applications we must solve this by other means. For $\lambda < \delta$ where δ is the smallest eigenvalue of the generalized eigenvalue problem $\boldsymbol{C}\boldsymbol{x} = \delta\boldsymbol{P}\boldsymbol{x}$, the solution will be unique if and only if the secular equation $\boldsymbol{\alpha}_n^\top \boldsymbol{P}\boldsymbol{\alpha}_n - s^2 = 0$ can be satisfied. Since the secular equation is strictly increasing for $\lambda \in]-\infty, \delta)$, we can instead perform a binary search in this range, in order to saturate $\boldsymbol{\alpha}_n^\top \boldsymbol{P}\boldsymbol{\alpha}_n = s^2$ with a sufficiently high precision. For more details we refer to Walder et al. (2010) and Gander et al. (1989).

2.2. The pre-image problem

Given a basis parameterized by a set of $\boldsymbol{\alpha}$'s determined by either standard kernel PCA or semi-supervised kernel PCA as described in the previous section, we are now interested in projecting a φ-mapped test point onto a principal subspace. For denoising applications we are interested in the projection onto the signal manifold, defined as a subspace of the RKHS spanned by the leading principal components. From the Representer Theorem, the projection of a feature space mapped test point onto the *n'th* principal component is

$$\beta_n(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_{ni} k_c(\boldsymbol{x}, \boldsymbol{x}_i) \tag{18}$$

where k_c is the centered kernel. The projection of $\varphi(\boldsymbol{x})$ onto the subspace spanned by the first q components will be denoted $P_q\varphi(\boldsymbol{x})$ and are given by

$$P_q\varphi(\boldsymbol{x}) = \sum_{n=1}^{q} \beta_n \sum_{i=1}^{N} \alpha_{ni} \varphi_c(\boldsymbol{x}_i) + \bar{\varphi} \tag{19}$$

where $\bar{\varphi} = \frac{1}{N}\sum_{i=1}^{N} \varphi(\boldsymbol{x}_i)$ is the mean of the φ-mapped data points and $\varphi_c(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}_i) - \bar{\varphi}$ is the centered feature space mapping of $\boldsymbol{x}$ (Schölkopf et al., 1998).

For denoising purposes it is of interest to reconstruct a data point in input space that corresponds to a specific linearly denoised point in feature space,

hence, applying the inverse map of φ. Thus, we are interested in finding a point $\boldsymbol{z} \in \mathcal{X}$ such that $\varphi(\boldsymbol{z}) = P_q\varphi(\boldsymbol{x})$ and we will call $\boldsymbol{z}$ the pre-image of $P_q\varphi(\boldsymbol{x})$.

The standard pre-image problem of reconstructing kernel PCA projections have been faced in a variety of ways, most of which are limited to a specific choice of kernel embedding (see e.g., Mika et al. (1999); Kwok and Tsang (2003); Dambreville et al. (2006); Bakir et al. (2004a)).

We follow the original work by Mika et al. (1999) and relax the problem to that of finding an approximate pre-image, i.e., a point in input space which maps into a point in feature space "as close as possible" to $P_q\varphi(\boldsymbol{x})$. To implement this search we seek to minimize the distance in the RKHS between $\varphi(\boldsymbol{z})$ and $P_q\varphi(\boldsymbol{x})$ with respect to $\boldsymbol{z}$. Thus, we use a quadratic objective function, which can be simplified as

$$\begin{aligned}\rho &= ||\varphi(\boldsymbol{z}) - P_q\varphi(\boldsymbol{x})||^2 \\ &= k(\boldsymbol{z},\boldsymbol{z}) - 2\sum_{n=1}^{N}\xi_n k(\boldsymbol{z},\boldsymbol{x}_n) + \Omega \end{aligned} \tag{20}$$

where all the $\boldsymbol{z}$-independent terms are collected in Ω, and $\xi_n = \tilde{\xi}_n + \frac{1}{N}(1 - \sum_{j=1}^{N}\tilde{\xi}_j)$, with $\tilde{\xi}_n = \sum_{i=1}^{q}\beta_i\alpha_{in}$.

In extrema, the derivative with respect to $\boldsymbol{z}$ is zero, which leads to the following fixed-point iteration for Gaussian kernels (Mika et al., 1999)

$$\begin{aligned}\boldsymbol{z}_{t+1} &= \frac{\sum_{n=1}^{N}\xi n \exp(-\gamma||\boldsymbol{z}_t - \boldsymbol{x}_n||^2)\boldsymbol{x}_n}{\sum_{n=1}^{N}\xi n \exp(-\gamma||\boldsymbol{z}_t - \boldsymbol{x}_n||^2)} \\ &= \frac{[\boldsymbol{\xi} \circ \boldsymbol{k}_{\boldsymbol{z}_t}]^\top \boldsymbol{X}}{\boldsymbol{\xi}^\top \boldsymbol{k}_{\boldsymbol{z}_t}} \end{aligned} \tag{21}$$

The cost in Equation (20) may be highly multi modal, leading to a nonlinear optimization problem, and hence the fixed-point iteration scheme can suffer from convergence to local minima. This typically implies sensitivity to the initial point $\boldsymbol{z}$ and can lead to instability of the denoising solution, see e.g., Abrahamsen and Hansen (2011).

Similarly, we now seek a fixed-point iteration for determining the pre-image when using the Graph kernel, $\tilde{k}$, as defined in Equation (1). When updating the pre-image estimate we will for simplicity assume that the pre-image itself is not part of $\boldsymbol{K}$. Thereby, we avoid the inversion of $(\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})^{-1}$

at every iteration that scales cubically. The effects of this relaxation will be reduced if the manifold is well defined by the training samples. By letting $\boldsymbol{M} := (\boldsymbol{I} + \boldsymbol{L}\boldsymbol{K})^{-1}\boldsymbol{L}$, the Graph kernel simplifies to $\tilde{k}(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{k}_x^\top \boldsymbol{M} \boldsymbol{k}_y$.

We now expand the $\boldsymbol{z}$ dependent terms of the cost function in Eq. (20) for this kernel

$$\begin{aligned}\rho &= \tilde{k}(\boldsymbol{z}, \boldsymbol{z}) - 2\sum_{n=1}^{N} \xi_n \tilde{k}(\boldsymbol{z}, \boldsymbol{s}_n) \qquad (22)\\ &= \exp(-\gamma||\boldsymbol{z} - \boldsymbol{z}||^2)\\ &\quad - \sum_{i,j=1}^{N} \exp(-\gamma||\boldsymbol{z} - \boldsymbol{s}_i||^2) M_{ij} \exp(-\gamma||\boldsymbol{z} - \boldsymbol{s}_j||^2)\\ &\quad - 2\Big[\sum_{n=1}^{N} \xi_n \big[\exp(-\gamma||\boldsymbol{z} - \boldsymbol{s}_n||^2) -\\ &\quad \sum_{i,j=1}^{N} \exp(-\gamma||\boldsymbol{z} - \boldsymbol{s}_i||^2) M_{ij} \exp(-\gamma||\boldsymbol{s}_n - \boldsymbol{s}_j||^2)\big]\Big] \qquad (23)\end{aligned}$$

Again the minima of Equation (23) are among points in which the derivative with respect to $\boldsymbol{z}$ is zero

$$\begin{aligned}&\sum_{i,j=1}^{N} \Big[M_{ij}(2\boldsymbol{z} - \boldsymbol{s}_i - \boldsymbol{s}_j) \exp(-\gamma(||\boldsymbol{z} - \boldsymbol{s}_i||^2 + ||\boldsymbol{z} - \boldsymbol{s}_j||^2))\\ &\quad - \Big[\sum_{n=1}^{N} \xi_n \big[-2(\boldsymbol{z} - \boldsymbol{s}_n) \exp(-\gamma||\boldsymbol{z} - \boldsymbol{s}_n||^2)\\ &\quad + \sum_{i,j=1}^{N} 2M_{ij}(\boldsymbol{z} - \boldsymbol{s}_i + \boldsymbol{s}_n - \boldsymbol{s}_j) \cdot\\ &\quad \exp(-\gamma(||\boldsymbol{z} - \boldsymbol{s}_i||^2 + ||\boldsymbol{s}_n - \boldsymbol{s}_j||^2))\big]\Big]\Big] = \boldsymbol{0} \qquad (24)\end{aligned}$$

Hence, we arrive at the following fixed-point iteration scheme for the Graph

kernel

$$\boldsymbol{z}_{t+1} = \frac{\left[(\boldsymbol{M} \circ (\boldsymbol{k}_{\boldsymbol{z}_t}\boldsymbol{k}_{\boldsymbol{z}_t}^\top - \boldsymbol{k}_{\boldsymbol{z}_t}(\boldsymbol{K}\boldsymbol{\xi})^\top - (\boldsymbol{K}\boldsymbol{\xi})\boldsymbol{k}_{\boldsymbol{z}_t}^\top))\mathbf{1}\right]^\top \boldsymbol{X}}{(\boldsymbol{k}_{\boldsymbol{z}_t}^\top \boldsymbol{M} + \boldsymbol{\xi}^\top - 2\boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{M})\boldsymbol{k}_{\boldsymbol{z}_t}} + \frac{\left[\boldsymbol{M} \circ \boldsymbol{\xi} \circ \boldsymbol{k}_{\boldsymbol{z}_t}\right]^\top \boldsymbol{X}}{(\boldsymbol{k}_{\boldsymbol{z}_t}^\top \boldsymbol{M} + \boldsymbol{\xi}^\top - 2\boldsymbol{\xi}^\top \boldsymbol{K}\boldsymbol{M})\boldsymbol{k}_{\boldsymbol{z}_t}} \tag{25}$$

3. Experimental results

In the following we evaluate the performance of denoising by semi-supervised kernel PCA on two data sets. To get some insights on the properties of the proposed methods we design a two-dimensional two-class problem with non-linear-separable manifolds by two intertwined spirals. Furthermore, we test the performance on a subset of The Amsterdam Library of Object Images (ALOI) database of images (Geusebroek et al., 2005).

3.1. Simulated data

To investigate the denoising performance of the proposed methods we construct a simple two class two dimensional synthetic data set as shown in the left panel of Figure 2. The data consists of two noisy entangled spirals where a random subset of the observations have label information.

For the experiments we retain 3 principal components and measure the quality of the denoising scheme by the mean squared error (MSE) of the reconstruction of a test set. Initially, we investigate the performance for varying signal-to-noise ratios by adding i.i.d. Gaussian noise to the data with zero mean and variance, σ^2. For all experiments we fixed the parameters of the semi-supervised model to $s = 10$ and $c = 2$.

We use 300 observations for training of which the label is known for 50 randomly chosen points from each class. The test set contains 100 unlabeled test points. The kernel-parameter is fixed to $\gamma = 5$. The results are summarized in the right panel of Figure 2, where we show error bars on the MSE as a function of the standard deviation of the Gaussian noise. It is evident that the semi-supervised reconstructions outperform their unsupervised counterparts for both the Gaussian and the Graph kernel for all noise levels. Furthermore, using the Graph kernel clearly leads to a better reconstruction measured by a lower MSE indicating a more descriptive manifold representation.

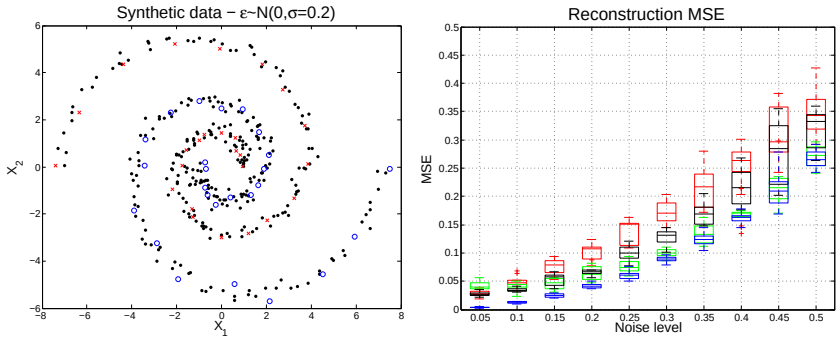


Figure 2: The left panel ilustrates the synthetic data used for our experiments. Colored samples are labeled whereas black samples are unlabeled. The right panel shows the mean squared error as a function of the standard deviation of the added Gaussian noise, σ. The green bars are kernel PCA with the Graph kernel, while the blue bars are semi-supervised kernel PCA with Graph kernel. Similarly, the red bars are kernel PCA with the Gaussian kernel, while the black bars are semi-supervised kernel PCA with the Gaussian kernel. Incorporating label information by the Graph kernel is seen to outperform the Gaussian kernel, and for both choices of kernel, semi-supervised learning is found to improve performance.

In order to investigate how much we learn from the unlabeled versus labeled samples we generate learning curves by fixing the noise level to $\sigma = 0.15$ and vary the number of observations used to learn the manifold structure. We learn the manifold fully supervised (with all labels known), unsupervised (standard kernel PCA), and semi-supervised (1/3 of the labels known) and compare the MSE of the pre-image reconstructions. The results are summarized in Figure 3, where the left panel shows the results achieved using the Gaussian kernel, and the right panel shows the results using the Graph kernel. As expected, for a fixed training set size fully supervised learning is preferable while completely unsupervised learning performs the worst. This tendency is less clear when only few samples are available. For the Graph kernel having 1/3 of the labels yields results comparable to knowing all labels. This is due to correct label propagation since the manifold assumption holds. It is important to notice how adding unlabeled samples significantly lower the MSE for both kernels. This is evident by comparing, e.g., the MSE for the supervised methods for $N = 100$ with the MSE achieved using the semi-supervised scheme for $N = 300$ (i.e, 100 labeled samples and 200 unla-

beled). For both kernel functions adding unlabeled observations leads to a significant lower MSE.

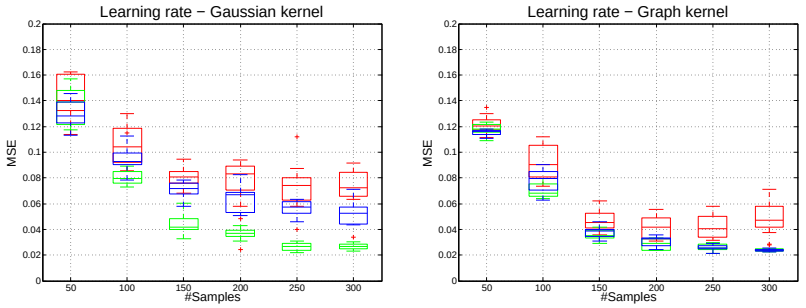


Figure 3: Illustrates the learning curves (mean squared error as a function of the training set size) for the synthetic data set depicted in Figure 2. The left panel shows the learning rates for the Gaussian kernel, while the right panel shows the learning rates for Graph kernel. The red bars are usual kernel PCA, i.e., unsupervised. Green bars: supervised kernel PCA, i.e., all training samples are labeled. Blue bars: semi-supervised kernel PCA with 1/3 of the training samples are labeled.

3.2. Amsterdam Library of Object Images

The Amsterdam Library of Object Images is a collection of images of 1000 objects that have been recorded for scientific purposes (Geusebroek et al., 2005). We consider a subset of 15 objects from the view point data set from this collection, where the view point is shifted in steps of 5° yielding a total of 72 images of each object. We treat each object as a class and assume to have 5 labeled samples (and 67 unlabeled) per class. Due to space constraints, we limit this section to a comparison between the unsupervised Gaussian kernel PCA approach and the semi-supervised Graph kernel PCA approach, as the two other combinations have shown to fall in-between the performance of the these two methodologies.

We construct a denoising problem by randomly adding two images from the database. The intensity of one of the images will be half of the intensity of the other, and the goal is to reconstruct the dominant image. For the semi-supervised methods we assume that the class label of the test image is known. Knowing the label of the test sample is justified by the fact that we are not focusing on classification, but merely aiming to incorporate side information for improved denoising. In case the test label was unknown an

initial classification step can be performed using the semi-supervised kernel PCA basis, since each leading eigenvector can be interpreted as an one-vs-rest classifier.

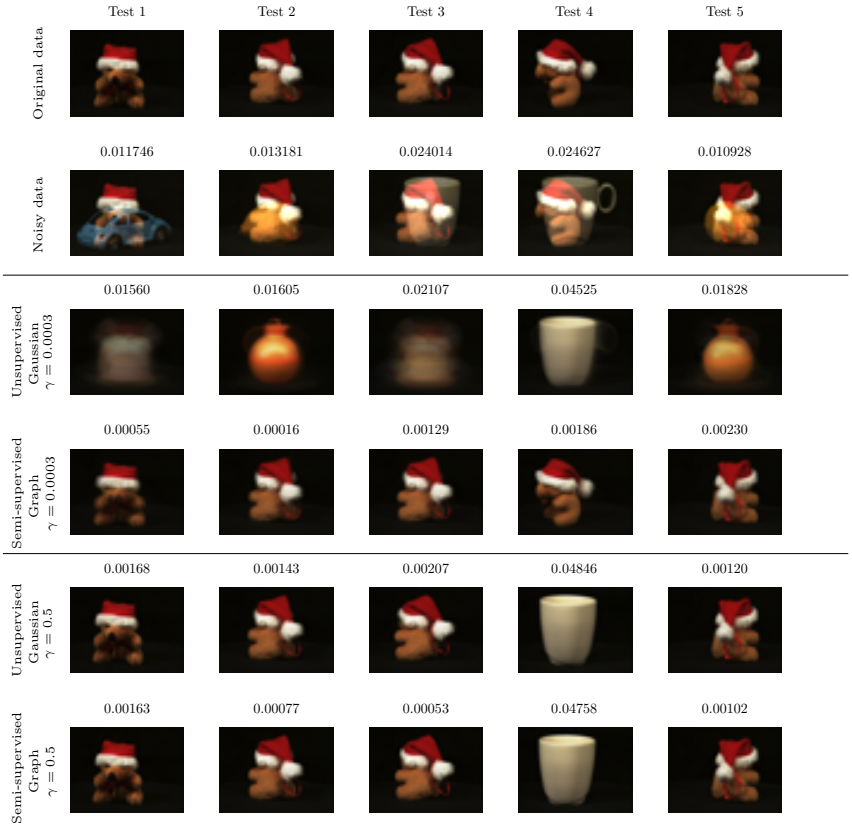


Figure 4: Examples of denoised images form the ALOI database. The first row shows the original test images that we seek to reconstruct, while the second row shows the constructed noisy test images. Above each image in row 2-6 are shown the MSE with respect to the original test image. The third and fourth row shows denoised images based on respectively unsupervised Gaussian and semi-supervised Graph kernel PCA; for both methods the kernel width has been fixed to $\gamma = 0.0204$ (leftmost point in Figure 5), and both utilize $q = 10$ PC's for the denosing task. The final two rows shows similar results but these are based on a more nonlinear kernel, where the parameter was fixed to $\gamma = 0.5$.

Figure 4 shows examples of the results using our proposed methods. For all experiments, we retain $q = 10$ PC's and for the semi-supervised model we set $s = 10$ and $c = 2$ as for the synthetic data. The top panel shows the original dominant test image, whereas the second panel shows the artificially constructed "noisy" image. The remaining panels show the denoised reconstructions for two choices of γ using unsupervised Gaussian kernel PCA and semi-supervised Graph kernel PCA respectively. The MSE with respect to the original test image is given above each image. It is evident both visually and from the MSE that the semi-supervised approach yields better results in all cases and it is more robust to the choice of kernel width, and aligns better with the original image (note the slight rotations of the reconstructions). For the most linear choice of kernel as measured by the kernel width, using the unsupervised Gaussian kernel is seen to fail in all cases while the semi-supervised version reconstruct meaningful images. For the more non-linear embedding the two reconstructions visually appear to be similar, however in terms of the MSE the semi-supervised kernel PCA is found to still be slightly superior.

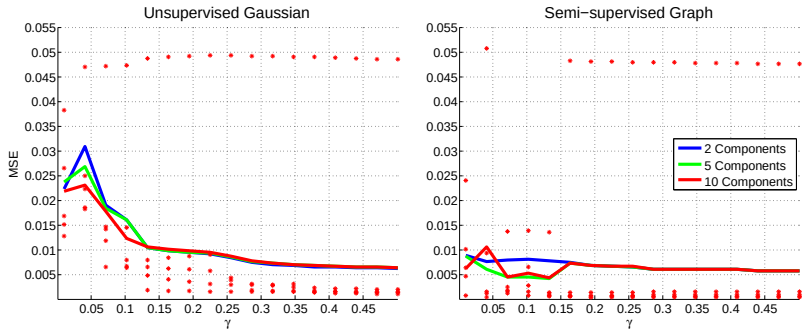


Figure 5: Shows the mean squared error as a function of the kernel width using a varying number of PCs to describe the manifold. Each point correspond to the MSE of one of the images in Figure 4 while the solid lines are the mean across all images. By comparing the right and left plots, it can be seen that the semi-supervised approach leads to better performance across all values of γ and in particular in the very linear regime. For both methods it is evident that the MSE is minimal for very large values of γ and hence very non-linear kernel embeddings, thereby suggesting that the optimal reconstruction is close a 1-nearest-neighbor approach.

Figure 5 shows the MSE as a function of the non-linearity of the kernel

embedding. We find that a nearest neighbor reconstruction (large γ) is close to optimal measured by the MSE as seen in Figure 5. This can be explained by the fine angular sampling within each class and by the complex nature of the signal manifold relative to the low number of PC's retained. However, we emphasize that the MSE for the Graph kernel has an significantly lower envelope than the other method, and that good results are obtained for a much broader range of kernel width parameters, making the Graph kernel easier to deploy in practice.

For the unsupervised Gaussian kernel, the poor performance in the linear regime can be explained as the consequence of the recovered leading principal components being unrelated to the denoising task of interest, thereby resulting in a high MSE. However, it should be noted that the task related components are still "hidden" in the span of the eigenvectors of the kernel matrix as long as it is not rank deficient. The key to the success of the semi-supervised approach is that the labeled samples forces the leading principal components to align with the task relevant directions independently of the choice γ. Thereby making this approach much less sensitive to the kernel width.

4. Conclusions

We have proposed two variants for incorporating label information into kernel PCA denoising. By extending the work of Walder et al. (2010) we derived an iterative scheme for finding more than one basis vector, leading to a semi-supervised kernel PCA framework that extends to a multidimensional orthonormal basis biased towards the labeled data.

Additionally, we derived a fixed-point iteration for the pre-image problem for the Graph kernel introduced by Sindhwani et al. (2005) as another way of including label information in the kernel PCA denoising scheme.

Viability was proven on both simulated data and images from the ALOI database. The experiments validated that semi-supervised learning can yield a more descriptive representation of the signal manifold in kernel PCA, and thereby improve the denoising performance compared to classical unsupervised kernel PCA denoising.

References

Abrahamsen, T. J., Hansen, L. K., 2011. Regularized pre-image estimation for kernel pca de-noising: Input space regularization and sparse reconstruction. Journal of Signal Processing Systems.

Bakir, G. H., Weston, J., Schölkopf, B., 2004a. Learning to find pre-images. In: Advances in Neural Information Processing Systems 16. MIT Press, pp. 449–456.

Bakir, G. H., Zien, A., Tsuda, K., 2004b. Learning to find graph pre-images. In: Pattern Recognition. Vol. 3175 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 253–261.

Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2006. Semi-Supervised Learning. MIT Press.

Dambreville, S., Rathi, Y., Tannenbaum, A., 2006. Statistical shape analysis using kernel PCA. In: IS&T/SPIE Symposium on Electrical Imaging.

Gander, W., Golub, G. H., von Matt, U., 1989. A constrained eigenvalue problem. Linear Algebra and its Applications 114(115), 815 – 839.

Geusebroek, J. M., Burghouts, G. J., Smeulders, A. W. M., 2005. The amsterdam library of object images. International Journal of Computer Vision 61 (1), 103–112.

Golub, G. H., 1973. Some modified matrix eigenvalue problems. SIAM Review 15 (2), 318–334.

Joachims, T., 2003. Transductive learning via spectral graph partitioning. In: Proceedings of the International Conference on Machine Learning. pp. 290–297.

Kwok, J. T., Tsang, I. W., 2003. The pre-image problem in kernel methods. In: Proceedings of the International Conference on Machine Learning. pp. 408–415.

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., Rätsch, G., 1999. Kernel pca and de-noising in feature spaces. In: Advances in neural information processing systems II. MIT Press, pp. 536–542.

Schölkopf, B., Smola, A., Müller, K.-R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10 (5), 1299–1319.

Sindhwani, V., Niyogi, P., Belkin, M., 2005. Beyond the point cloud: from transductive to semi-supervised learning. In: Proceedings of the 22nd international conference on Machine learning. ACM, pp. 824–831.

Walder, C., Henao, R., Mørup, M., Hansen, L. K., 2010. Semi-supervised kernel pca. CoRR abs/1008.1398.

Appendix E

A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis

Trine Julie Abrahamsen and Lars Kai Hansen. A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis. *Journal of Machine Learning Research*, 12:2027–2044, 2011. Available from: `http://jmlr.csail.mit.edu/papers/v12/abrahamsen11a.html`.

Citations
This paper has been cited by [AH12, GMARH12, TMA+12, AGPCVH13, BHH+13, FWXT13, MFA+13, SZD13].

Journal of Machine Learning Research 12 (2011) 2027-2044 Submitted 1/11; Published 6/11

A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis

Trine Julie Abrahamsen TJAB@IMM.DTU.DK
Lars Kai Hansen LKH@IMM.DTU.DK
DTU Informatics
Technical University of Denmark
Richard Petersens Plads, 2800 Lyngby, Denmark

Editor: Manfred Opper

Abstract

Small sample high-dimensional principal component analysis (PCA) suffers from variance inflation and lack of generalizability. It has earlier been pointed out that a simple leave-one-out variance renormalization scheme can cure the problem. In this paper we generalize the cure in two directions: First, we propose a computationally less intensive approximate leave-one-out estimator, secondly, we show that variance inflation is also present in kernel principal component analysis (kPCA) and we provide a non-parametric renormalization scheme which can quite efficiently restore generalizability in kPCA. As for PCA our analysis also suggests a simplified approximate expression.

Keywords: PCA, kernel PCA, generalizability, variance renormalization

1. Introduction

While linear dimensionality reduction by principal component analysis (PCA) is a trusted machine learning workhorse, kernel based methods for *non-linear* dimensionality reduction are only starting to find application. We expect the use of non-linear dimensionality reduction to expand in many applications as recent research has shown that kernel principal component analysis (kPCA) can be expected to work well as a pre-processing device for pattern recognition (Braun et al., 2008). In the following we consider non-linear signal detection by kernel PCA followed by a linear discriminant classifier.

In spite of its conceptual simplicity and ubiquitous use, principal component learning in high dimensions is in fact highly non-trivial (see, e.g., Hoyle and Rattray, 2007; Kjems et al., 2001). In the physics literature much attention has been devoted to learnability phase transitions. In PCA there is a sharp transition as function of sample size from *no learning at all* to a regime where the projections become more and more accurate. In the transition regime where learning is still incomplete there is a mismatch between the test and training projections. In Kjems et al. (2001) it was shown that this can be interpreted as a case of *over-fitting* and leads to pronounced *variance inflation* in the training set projections and results in lack of generalization to test data as illustrated in Figure 1.

Variance inflation is of particular concern if PCA is used to reduce dimensionality prior to, for example, a classifier. When the data analytic pipeline is applied to test data the reduced variance of the PCA text projections can lead to significantly reduced performance. Fortunately, the bias can

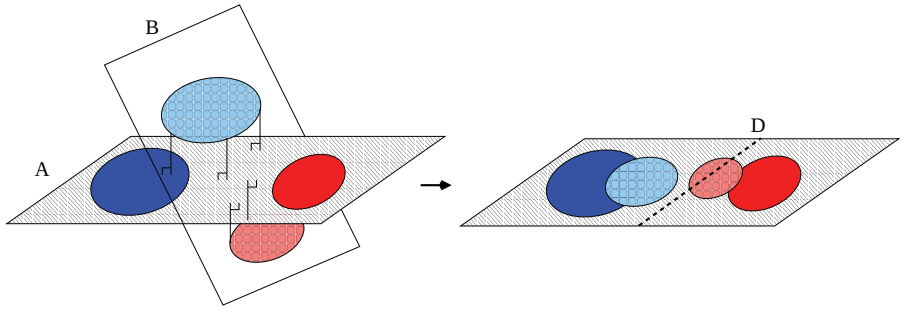


Figure 1: Illustration of the variance inflation problem in PCA. Because PCA maximizes variance, small data sets in high dimensions will be overfitted. When the PCA subspace (A) is applied to a test data set (B) the projected data will have smaller variance. This leads to lack of generalizability if the training data is used to train a classifier, say a linear discriminant (D). In Kjems et al. (2001) this problem was noted and it was shown that the necessary renormalization can be estimated in a leave-one-out procedure

be reduced effectively by a leave-one-out (LOO) scale renormalization of the PCA test projections to restore generalizability (Kjems et al., 2001). In this paper we pursue several extensions of this result. We give a straightforward geometric analysis of the projection problem that suggests a computationally less intensive approximate cure than the one originally proposed by Kjems et al. (2001). Next, we proceed to investigate the issue in the context of *kernel* based unsupervised dimensionality reduction. We show in both simulation and in real world data (USPS handwritten digits and functional MRI data) that variance inflation also happens in kPCA and basically for the same reasons as in PCA. We then provide an extension to the LOO procedure for kPCA which can cope with potential non-Gaussian distributions of the kPCA projections, and finally we propose a simplified approximate renormalization scheme.

2. Generalizability in PCA

The most complete theoretical picture of principal component learning is presented by Hoyle and Rattray (2007), which builds on and extends earlier work by, for example, Biehl and Mietzner (1994), Hoyle and Rattray (2004c), Johnstone (2001), Reimann et al. (1996), and Silverstein and Combettes (1992). Hoyle and Rattray (2007) consider a general PCA model with a multidimensional normal distributed signal that emerges from an isotropic noise background as the sample size increases. The stabilization of a given principal component happens at a given sample size and takes the form of a phase transition. For small sample sizes -below the phase transition point - the training set principal component eigenvectors are in completely random directions in space and there is no learning at all. Then, as the sample size increases, the first principal component stabilizes, and for even larger sample sizes the second, and so forth. Sharp transitions are strictly present only in a limit where both dimensionality and sample size are infinite with a finite ratio $\alpha = N/D$, but the theoretical results are very accurate at realistic dimensions as seen in Figure 2. The location of the first

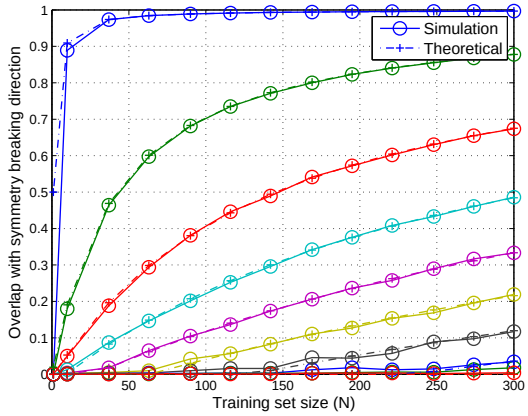


Figure 2: Phase transitions in PCA. Simulated data was created as $\boldsymbol{x} = \eta \boldsymbol{u} + \boldsymbol{\epsilon}$, with a normal distributed signal of unit strength $\eta \sim N(0,1)$, embedded in i.i.d. normal noise $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{1})$. In this simulated data set we show the phase transition like behavior of the overlap (the mean square of the projection) of the first PCA eigenvector and the signal direction $\boldsymbol{u}$. The input space has dimension $D = 1000$, and the curves are for 10 values of signal to noise within the interval $\sigma \in [0.01, 0.5]$. For a noise level of, for example, $\sigma = 0.17$ (black curves) there is a sharp transition both in the theoretical curve (dash/cross) and the experimental curve (full/circle) around $N = 120$ examples.

phase transition depends on the signal variance to noise variance ratio (SNR). The theoretical result provides a *mean bias* for a specific model, hence, cannot directly be used to restore generalizability in a given data set.

Now, what happens to the generalization performance of PCA in the noisy region? The PCA projections will be offset by different angles depending on how severe the given component is affected by the noise. Because of the bias the test projections will follow different probability laws than the training data, typically with much lower variance. Hence, if we train a classifier on the training projections the classifier will make additional errors on the test set as visualized in Figure 1.

In the case of PCA the subspace projections are uncorrelated, hence, it is meaningful to renormalize them independently. Assuming approximate normality, a simple affine transformation suffices. The scale factor is simply the ratio of the standard deviations of the training and test projections and can be estimated by a leave-one-out procedure (Kjems et al., 2001). However, since the LOO procedure involves the computation of N SVD's of an $(N-1) \times (N-1)$ matrix, it is of interest to find a simplified estimate.

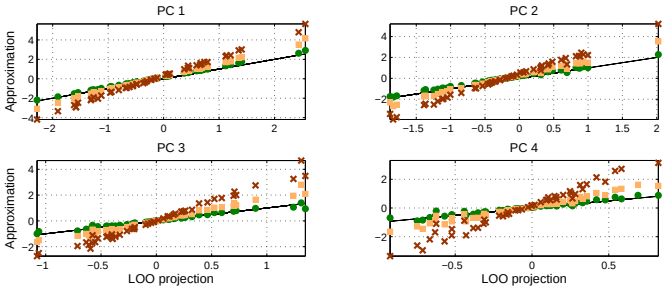


Figure 3: Approximating the leave-one-out (LOO) procedure. Here we simulate data with four normal independent signal components, $\boldsymbol{x} = \Sigma_{k=1}^{4} \eta_k \boldsymbol{u}_k + \boldsymbol{\epsilon}$ of strengths $(1.4, 1.2, 1.0, 0.8$, embedded in i.i.d. normal noise $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{1})$, with $\sigma = 0.2$. The dimension was $D = 2000$ and the sample size was $N = 50$. In the four panels we show the training set projections (red crosses), the projections corrected for the theoretical mean overlap (Hoyle and Rattray, 2007) (yellow squares) and the geometric approximation in Equation (1) (green dots) versus the exact LOO projections (black line).

Let $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be N training data points in a D dimensional input space $\mathcal{X}$ (see notation),[1] we consider the case $N \ll D$. The LOO step for the N'th point $\boldsymbol{x}_N$ concerns projecting onto the PCA eigenvectors derived from the subset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}\}$. Define the orthogonal and parallel components of the test point, $\boldsymbol{x}_N = \boldsymbol{x}_N^{\perp} + \boldsymbol{x}_N^{\parallel}$, relative to the subspace spanned by the training data. As the PCA eigenvectors with non-zero variance are all in the span of the training data we obtain

$$\boldsymbol{u}_{N-1,k}^T \cdot \boldsymbol{x}_N = \boldsymbol{u}_{N-1,k}^T \cdot \boldsymbol{x}_N^{\parallel} \, ,$$

where $\boldsymbol{u}_{N-1,k}$ is the k'th eigenvector of the LOO training set. Assuming that the changes in the PCA eigenvectors going from sample size N to $N-1$ are small, we can approximate the test projections as

$$\boldsymbol{u}_{N-1,k}^T \cdot \boldsymbol{x}_N = \boldsymbol{u}_{N-1,k}^T \cdot \boldsymbol{x}_N^{\parallel} \approx \boldsymbol{u}_{N,k}^T \cdot \boldsymbol{x}_N^{\parallel} \, , \tag{1}$$

where $\boldsymbol{u}_{N,k}$ is the k'th eigenvector on the full sample. The approximation introduces a small error of order $1/N$ as discussed in detail in the Appendix and further illustrated in a simulation data set in Figure 3. Note that the orthogonal projections $\boldsymbol{x}_N^{\parallel}$ of the N points may be calculated from the inverse matrix of the inner products of all data points, in N steps each of a cost scaling as N^2, thereby achieving a computational burden which scales as N^3 rather than the N^4 scaling for an exact LOO procedure proposed in Kjems et al. (2001).

1. Bold uppercase letters denote matrices, bold lowercase letters represent column vectors, and non-bold letters denote scalars. $\boldsymbol{a}_j$ denotes the *j'th* column of $\boldsymbol{A}$, while a_{ij} denotes the scalar in the *i'th* row and *j'th* column of $\boldsymbol{A}$. Finally $\mathbf{1}_{NN}$ is a $N \times N$ matrix of ones.

3. Renormalization Cure for Variance Inflation in kernel PCA

The statistical properties of kernel PCA have also been studied extensively by Blanchard et al. (2007), Hoyle and Rattray (2004a), Hoyle and Rattray (2004b), Mosci et al. (2007), Shawe-Taylor and Williams (2003) and Zwald and Blanchard (2006), but to our knowledge the geometry of generalization for kPCA has not been discussed in the extremely ill-posed case $N \ll D$.

To better understand the variance inflation problem in relation to kPCA let us recapitulate some basic aspects of this non-linear dimensional reduction technique.

Let $\mathcal{F}$ be the Reproducing Kernel Hilbert Space (RKHS) associated with the kernel function $k(\boldsymbol{x},\boldsymbol{x}') = \varphi(\boldsymbol{x})^T\varphi(\boldsymbol{x}')$, where $\varphi : \mathcal{X} \mapsto \mathcal{F}$ is a possibly non-linear map from the D-dimensional input space $\mathcal{X}$ to the high dimensional (possibly infinite) feature space $\mathcal{F}$. In kPCA the PCA step is carried out in the feature space, $\mathcal{F}$, mapped data (Schölkopf et al., 1998). However, as $\mathcal{F}$ can be infinite dimensional we first apply the kernel trick allowing us to work with the Gram matrix of inner products. Let $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ be N training data points in $\mathcal{X}$ and $\{\varphi(\boldsymbol{x}_1),\ldots,\varphi(\boldsymbol{x}_N)\}$ be the corresponding images in $\mathcal{F}$. The mean of the φ-mapped data points is denoted $\overline{\varphi}$ and the 'centered' images are given by $\tilde{\varphi}(\boldsymbol{x}) = \varphi(\boldsymbol{x}) - \overline{\varphi}$. The kPCA is performed by solving the eigenvalue problem $\widetilde{\boldsymbol{K}}\boldsymbol{\alpha}_i = \lambda_i\boldsymbol{\alpha}_i$ where the centered kernel matrix, $\widetilde{\boldsymbol{K}}$, is defined as

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \frac{1}{N}\mathbf{1}_{NN}\boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\mathbf{1}_{NN} + \frac{1}{N^2}\mathbf{1}_{NN}\boldsymbol{K}\mathbf{1}_{NN} \; . \tag{2}$$

The projection of a φ-mapped test point onto the *i'th* component is given by

$$\beta_i = \tilde{\varphi}(\boldsymbol{x})^T\boldsymbol{v}_i = \sum_{n=1}^{N}\alpha_{in}\tilde{\varphi}(\boldsymbol{x})^T\tilde{\varphi}(\boldsymbol{x}_n) = \sum_{n=1}^{N}\alpha_{in}\tilde{k}(\boldsymbol{x},\boldsymbol{x}_n) \; , \tag{3}$$

where $\boldsymbol{v}_i$ is the *i'th* eigenvector of the feature space covariance matrix and the $\boldsymbol{\alpha}_i$'s have been normalized. The centered kernel function can be found as $\tilde{k}(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{x},\boldsymbol{x}') - \frac{1}{N}\mathbf{1}_{1N}\boldsymbol{k}_{\boldsymbol{x}} - \frac{1}{N}\mathbf{1}_{1N}\boldsymbol{k}_{\boldsymbol{x}'} + \frac{1}{N^2}\mathbf{1}_{1N}\boldsymbol{K}\mathbf{1}_{N1}$, where $\boldsymbol{k}_{\boldsymbol{x}} = [k(\boldsymbol{x},\boldsymbol{x}_1),\ldots,k(\boldsymbol{x},\boldsymbol{x}_N)]^T$. The projection of $\varphi(\boldsymbol{x})$ onto the first q principal components will in be denoted $P_q(\boldsymbol{x})$.

In the following we focus on a Gaussian kernel of the form $k(\boldsymbol{x},\boldsymbol{x}') = \exp(-\frac{1}{c}||\boldsymbol{x}-\boldsymbol{x}'||^2)$, where c is the scale parameter controlling the non-linearity of the kernel map. By the centering operation, PCA is the obtained in the limit when $c \to \infty$. Thus for large values we expect variance inflation to be present due the reasons discussed above. What happens in the non-linear regime with a finite c? To answer this question we analyze the LOO scenario for kPCA.

Consider the squared distance $||\boldsymbol{x}_n - \boldsymbol{x}_N||^2$ in the exponent in the Gaussian kernel for some training set point $\boldsymbol{x}_n$ and a test point $\boldsymbol{x}_N$. If we split the test point in the orthogonal components as above with respect to the subspace spanned by the training set we obtain,

$$||\boldsymbol{x}_n - \boldsymbol{x}_N||^2 = ||\boldsymbol{x}_n - \boldsymbol{x}_N^{||}||^2 + ||\boldsymbol{x}_N^{\perp}||^2 \; .$$

Inserting this expression in the Gaussian kernel in Equation (3) it is seen that the test projection acquire a common factor $\exp\left(-\frac{1}{c}||\boldsymbol{x}_N^{\perp}||^2\right)$:

$$\beta_i(\boldsymbol{x}_N) = \sum_{n=1}^{N-1}\alpha_{in}\tilde{k}(\boldsymbol{x}_N,\boldsymbol{x}_n) = \exp\left(-\frac{1}{c}||\boldsymbol{x}_N^{\perp}||^2\right)\sum_{n=1}^{N-1}\alpha_{in}\tilde{k}(\boldsymbol{x}_N^{||},\boldsymbol{x}_n) \; ,$$

which can be arbitrary small for small values c, that is, in the non-linear regime.

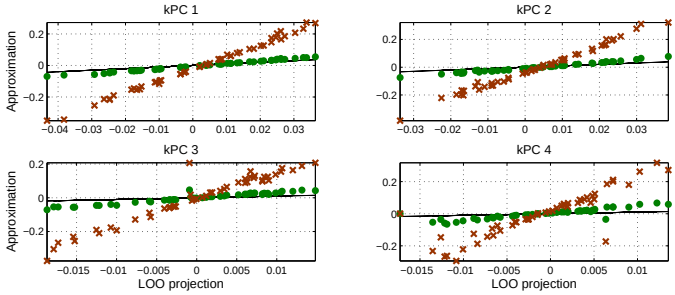


Figure 4: Approximating the leave-one-out (LOO) procedure for kPCA. We simulate a data set with four normal independent signal components, $\boldsymbol{x} = \Sigma_{k=1}^{4} \eta_k \boldsymbol{u}_k + \boldsymbol{\epsilon}$ of strengths $(1.4, 1.2, 1.0, 0.8$, embedded in i.i.d. normal noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$, with $\sigma = 0.2$. The dimension was chosen $D = 2000$ and the sample size was $N = 50$. In the four panels we show the four kPCA component's training set projections (red crosses), and the result of applying the point wise correction factor $\exp(\frac{1}{c}||\boldsymbol{x}_N^{\perp}||^2)$ for the lost orthogonal projection (green dots) versus the exact LOO kPCA test projections (black).

For a coordinate-wise LOO renormalization procedure we thus propose to compute N test projections by repeated kPCA on the $N-1$ sized sub training sets. However, compared to the PCA case we face two additional challenges, namely the potentially strongly non-Gaussian distributions and component dependencies.

To check for dependency we appeal to simple pairwise permutation test of significant mutual information measure (see, e.g., Moddemeijer, 1989). If the null hypothesis is rejected for a given set of components we cannot expect coordinate-wise renormalization to be effective. If, on the other hand, the kernel PCA projections pass the independence test we can proceed to renormalize the components individually. In the following we will assume that a coordinate-wise approach is acceptable. First, as a simple approximation to the full LOO we consider adjusting for the common scaling factor due to the lost orthogonal projection. This may indeed provide for viable approximation as seen in Figure 4.

To address the second challenge, namely the potential non-normality we propose to generalize the affine scaling method of Kjems et al. (2001) by a non-parametric procedure. Assume that there exists a monotonic transformation between the N training and N LOO test set projections. The problem of calibrating for an unknown monotone transformation is a common operation in image processing, and is used, for example, to transform the gray scale of an image in order to standardize the pixel histogram (Gonzalez and Wintz, 1977). Equalizing two equal sized samples, simply involves sorting both and assigning the sorted test projections the sorted values of the training projections, this procedure is easily seen to equalize the histograms without changing the level sets (relative ordering) of the LOO test projections. In Figure 5 a simple 1-dimensional data set is used to illustrate the equalization procedure. The training set clearly contains two classes. However, due to variance inflation (induced by, for example, kernel PCA) the test set does not follow the same

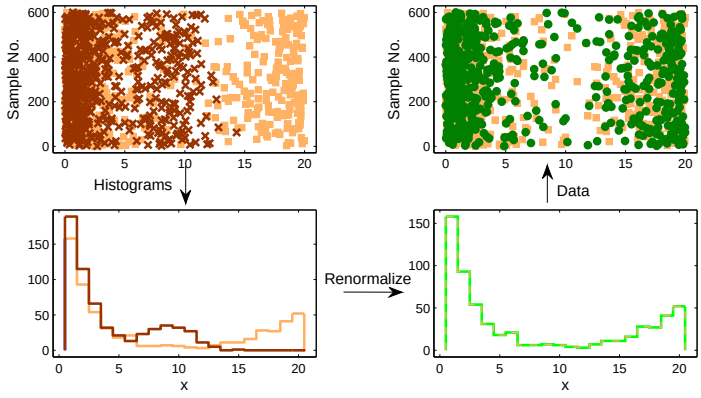


Figure 5: Illustration of renormalization by histogram equalization. The left panel shows the training set (yellow squares) and original test set (red crosses) and their respective histograms. The histograms are then equalized as seen in the right panel, where the green dots are the renormalized test data. The renormalization clearly restores the variation of the test set.

distribution, and may potentially lead to a high misclassification rate. The right panel of the figure shows how histogram equalization restores generalizability.

Technically, the transformation may be described as follows. Let $H(f)$ be the cumulative distribution of values f of a given kPCA projection of the training set. Let the test set projections on the same component for N_{test} samples take values $g(m)$. Let $I(m)$ be the index of sample m in a sorted list of the test set values. Then the renormalized value of the test projection m is

$$\widetilde{g(m)} = H^{-1}(I(m)/N_{\text{test}}) \ .$$

The test set projections can be obtained by the simple relation

$$\widetilde{g(m)} = f_{\text{sort}}(I(m)) \ , \tag{4}$$

where f_{sort} is the sorted list of training set projections. The algorithm for approximate renormalization is summarized in Algorithm 1.[2]

4. Evaluation of the Proposed Cure in Classification Problems

In the following we evaluate the non-parametric exact LOO correction scheme when kPCA is used as a dimensional reduction step in simulated and real classification data sets.

2. We thank the reviewers for pointing out that while non-normality is expected in the case of kPCA, non-normality may also appear in PCA calling for application of the proposed non-parametric renormalization scheme in this case.

Algorithm 1 Approximate renormalization in kernel PCA

Require: $\boldsymbol{X}_{tr}$ and $\boldsymbol{X}_{te}$to be $N_{tr} \times D$ and $N_{te} \times D$ respectively
  Compute $\widetilde{\boldsymbol{K}}_{tr}$ using Equation (2) and find the eigenvectors, $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q$
  for $i = 1$ to N_{tr} **do**
    $\boldsymbol{f}_{tr}^{i,:} \leftarrow P_q(\boldsymbol{x}_{tr}^{i,:}) = \tilde{\boldsymbol{k}}_{x_i}^T \boldsymbol{\alpha}^{i:q}$ {see Equation (3)}
  end for
  for $j = 1$ to N_{te} **do**
    $\boldsymbol{f}_{te}^{j,:} \leftarrow P_q(\boldsymbol{x}_{te}^{j,:}) = \tilde{\boldsymbol{k}}_{x_j}^T \boldsymbol{\alpha}^{i:q}$ {see Equation (3)}
  end for
  for $d = 1$ to q **do**
    $[\boldsymbol{f}_{sort}, \ ] \leftarrow \text{sort}(\boldsymbol{f}_{tr}^{:,d})$ {ascending order}
    $[\ , I] \leftarrow \text{sort}(\boldsymbol{f}_{te}^{:,d})$ {ascending order}
    if $N_{tr} = N_{te}$ **then**
      $\boldsymbol{h} \leftarrow \boldsymbol{f}_{sort}$
    else $\{N_{tr} \neq N_{te}\}$
      $\boldsymbol{h} \leftarrow \text{spline}\big([1 : N_{tr}], \boldsymbol{f}_{sort}, \text{linspace}(1, N_{tr}, N_{te})\big)$ {interpolate to create N_{te} values of $\boldsymbol{f}_{sort}$ in the interval $[1 : N_{tr}]$}
    end if
    for $n = 1$ to N_{te} **do**
      $\tilde{\boldsymbol{g}}_{te}^{I(n),d} \leftarrow \boldsymbol{h}^{n,d}$ {renormalized test data in the principal subspace, see Equation (4)}
    end for
  end for

4.1 Simulated Data

To get some insight into the non-linear regime, we design a synthetic data set containing two 2-dimensional semi-circular clusters which cannot be separated linearly (cf., Jenssen et al., 2006). Gaussian noise is added to one of the clusters, and the data is further embedded in 1000 'noise dimensions'. The basis is changed so that the 2D signal space occupies a general position. The noise is as earlier assumed i.i.d. with variance σ^2. The assignment variable is $t = 0, 1$, and in the experiments the data set is assumed unbalanced with $p(t = 0) = 0.6$.

In Figure 6 we show in the left panel a linear discriminant trained on the training set projections in a data set of $N = 500$ in $D = 1000$ dimensions. The role of the non-linearity as controlled by the parameter c in the Gaussian kernel is investigated in Figure 7 for a simulation setup similar to Figure 6. As seen the inflation problem dramatically amplifies as non-linearity increases. Finally, Figure 8 shows how renormalization improves the learning curve for the same problem.

4.2 USPS Handwritten Digit Data

The USPS handwritten digit benchmark data set is often used to illustrate unsupervised and supervised kernel methods. The USPS data set consists of $D = 16 \times 16 = 256$ pixels handwritten digits.[3] For each digit we randomly chose 10 examples for training and another 10 examples for testing. The scale was chosen as the 5th percentile of the mutual distances of the data points leading to $c \approx 120$,

3. The USPS data set is described by Hull (1994) and can be downloaded from `www.kernel-machines.org`.

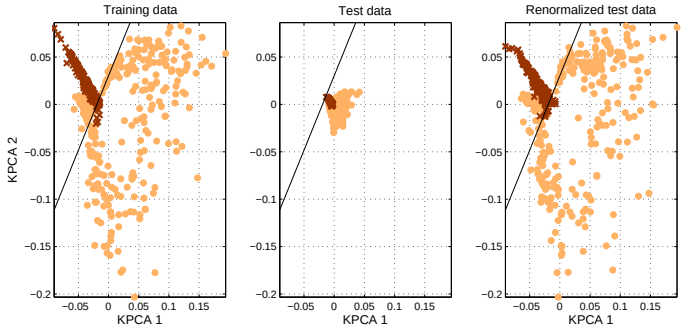


Figure 6: An unbalanced two cluster data set showing a pronounced variance inflation problem in the projections of the test data in the middle panel. In the right panel we have applied the cure based on non-parametric renormalization to equalize training and test projections using histogram equalization. The linear discriminant performs close to the optimal Bayes rate after non-parametric renormalization. The sample size is $N = 500$ in $D = 1000$ dimensions and the SNR is 10. The training error rate is 0.002 while the uncorrected test error rate is 0.4. Renormalization reduces the test error to 0.002.

and the number of principal components was chosen so 85% of the variance was contained in the principal subspace leading to around $q = 57$ PCs to be included.

The first step is to submit the data to the mutual information permutation test. For every pair of principal components a permutation test with 1000 permutations was performed in order to test the null hypothesis of the two given components being independent. Using a $\rho = 0.05$ significance level, we find that the null hypothesis can only be rejected for approximately 2% of the principal component pairs when not using Bonferroni correction. The combinations for which the null hypothesis can be rejected are equally distributed across the principal components. Since the expected number of rejected tests at the given confidence level is 5%, hence, we can safely proceed with the coordinate-wise renormalization process.

In the q dimensional principal subspace the projections of the test set are renormalized to follow the training set histogram. We chose in these experiments for demonstration to classify digit 8 versus the rest. A linear discriminant classifier was trained on the kernel PCA projections of the training set, and the classification error was found using both the conventional kernel PCA projections of the test set and their renormalized counterparts. In order to compare the two methods, the procedure was repeated 300 times using random training and test sets. While classification based on the conventional projections resulted in a mean classification error rate ($\pm$ 1 std) of 0.06 ± 0.01, using the renormalized projections lowered the error rate to 0.05 ± 0.02. A paired t-test showed that this reduction is highly significant ($p = 2.0875 \cdot 10^{-11}$).

Figure 9 shows an example of the projections before and after renormalization. The axis are fixed across the two methods. The top row clearly illustrates the inflation problem for conventional kPCA. Furthermore, due to the imbalanced nature of the data set, the inflation causes a high misclas-

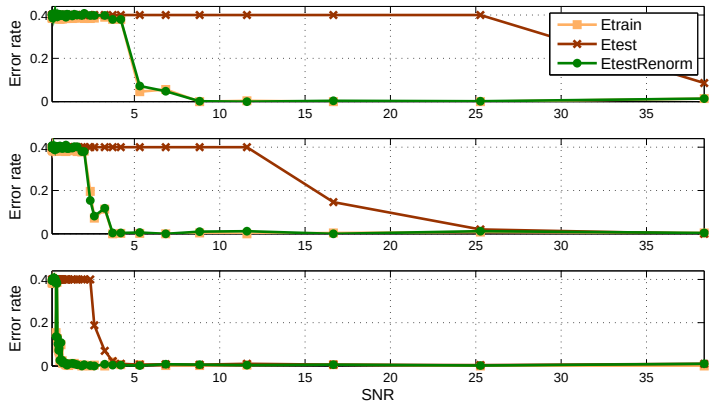


Figure 7: The role of non-linearity on the variance inflation problem. We carry out three experiments at different values of the Gaussian kernel scale parameter (top to bottom: $c = 0.05$, $c = 0.1$, $c = 0.5$). We show classification errors as a function of SNR. The linear discriminant performs close to the optimal Bayes rate after the renormalization operation in all cases, while the un-renormalized systems suffers from poor generalizability. The sample size is $N = 500$ and the number of dimensions is $D = 1000$.

sification rate. The bottom row illustrates how renormalization overcomes the distortions induced by the variance inflation. The discriminant line is seen to separate the two classes appropriately.

To gain a better understanding of how the variance inflation and quality of the renormalization are effected by noise, we added Gaussian noise ($\mathcal{N}(0,\sigma_e^2)$) with $\sigma_e \in [0,5]$. For every noise level, 300 random training and test sets where drawn as explained above and kPCA was performed. Once again our goal was to classify digit 8 versus the rest by a linear classifier in the principal subspace. The results are summarized in Figure 10 where we show the error rate before and after renormalization as well as the result based on renormalizing according to the leave-one-out error. In the last case, the N projections determined from leave-one-out cross validation (LOOCV) are renormalized to follow the entire training set histogram. Renormalization is then only applied to the test set when this renormalized LOOCV error is less than the estimated baseline error. In the right panel of Figure 10 it is seen how renormalizing the projections leads to a much improved classifier as long as the SNR is 'reasonable'. Even when $\sigma_e = 0$ there is some inherent noise in the data, which explains why renormalization still improves the classification. As σ_e reaches 1 it is no longer possible to identify the digits by visual inspection, and classification becomes increasingly difficult.

The left panel of Figure 10 shows how the conventional error rate converges to the baseline of 0.1 (misclassifying all digits 8), for high noise levels. Basically, increasing the noise result in a more skewed test set subspace in relation to the subspace spanned by the training set (see Figure 1). At a given threshold this causes all the projections to lie on the same side of the discrimination function due to the imbalanced composition, leading to a misclassifications rate of $1/10$. As the idea of renormalization by histogram equalization is to restore the variation in the test set, this be-

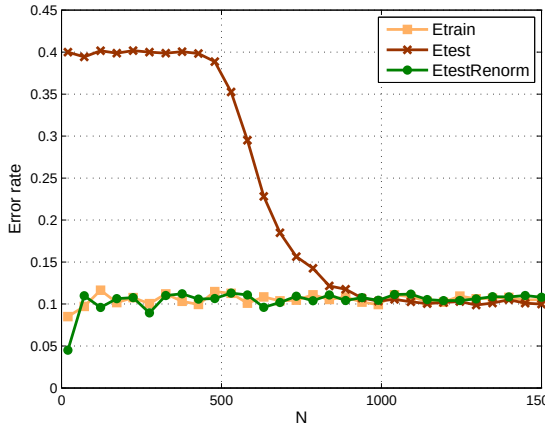


Figure 8: Classification error learning curves for the two semicircular clusters in i.i.d. noise setup. The signal to noise ratio was $SNR = 60$. The linear discriminant performs close to the optimal Bayes rate after the renormalization operation in all cases, while the conventional system suffers from poor generalizability, and requires about ten times as many examples to reach the same error level as the renormalized classifier. The experiment was carried out with $D = 2000$.

havior is naturally not encountered for the renormalized projections. Instead, as the SNR decreases, renormalization increases the error rate, as the test set observations are forced to be distributed on both sides of the discrimination line - which leads to many misclassifications when the signal is suppressed by the noise. However, using LOOCV based renormalization prevents the error rate from blowing up while at the same time improving the classification in the more sensible SNR regime as compared to conventional kPCA.

4.3 Functional MRI Data

As a second high dimensional real data example, functional magnetic resonance imaging (fMRI) data was used to illustrate the effect of renormalization. The fMRI data set was acquired by Dr. Egill Rostrup at Hvidovre Hospital on a 1.5 T Magnetom Vision MR scanner. The scanning sequence was a 2D gradient echo EPI (T2- weighted) with 66 ms echo time and 50° RF flip angle. The images were acquired with a matrix of $D = 128 \times 128 = 16,384$ pixels, with FOV of 230 mm, and 10 mm slice thickness, in a para-axial orientation parallel to the calcarine sulcus. The visual paradigm consisted of a rest period of 20 sec of darkness using a light fixation dot, followed by 10 sec of full-field checkerboard reversing at 8 Hz, and ending with 20 sec of rest (darkness). In total, 150 images were acquired in 50 sec, corresponding to a period of approximately 330 msec per image. The experiment was repeated in 10 separate runs containing 150 images each. In order to reduce saturation effects, the first 29 images were discarded, leaving 121 images for each run. We use a

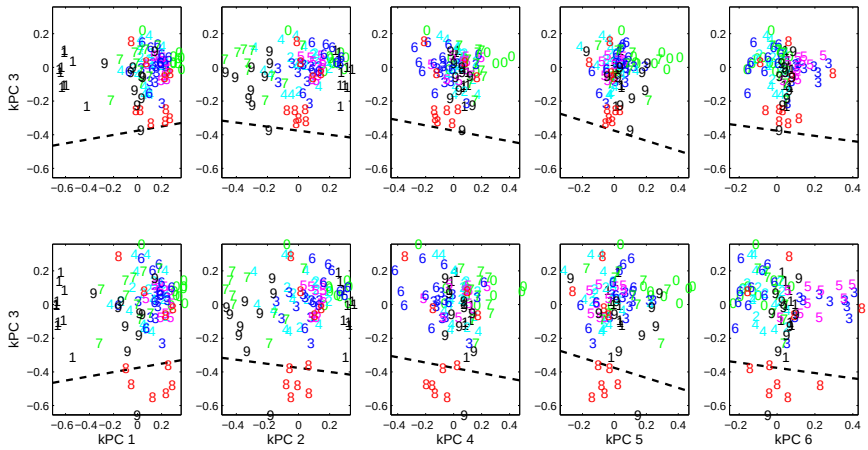


Figure 9: USPS handwritten digits test set projections. The top row shows the conventional projections, while the bottom row shows the projections after renormalization. In this example the third kPC carries a large part of the signal, and hence this component is shown versus the other five first PCs. The variance reduction and the consequent shift is evident from the top row. The dashed line indicates the linear discriminant function for classifying digit 8 vs the rest.

simple on-off activation reference function for supervision of the classifier. The reference function is off-set by 4 seconds to emulate the hemodynamic delay.

The data set is split in two equal sized subsets: Five runs for training and five runs for testing. As the test and training data are independent, the test error estimate is an unbiased estimator of performance. The scale of the Gaussian kernel was chosen as the 5th percentile of the mutual distances leading to $c \approx 15000$, while the dimension of the principal subspace is chosen as $q = 20$.

Again the principal components are tested for independence by a mutual information permutation test. Using 1000 permutations and a $\rho = 0.05$ significance level, we find that the null hypothesis is rejected for approximately 1% of the principal component pairs.

Similar to the handwritten digit data we perform linear classification in the kernel principal subspace. This was repeated 300 times using random splits for different noise levels. The results are summarized in Figure 11. Again renormalization is seen to decrease the error rate significantly, while the LOOCV based scheme furthermore prevents the increase in error rate for high noise levels (low SNR).

Figure 12 shows the projection of the data onto the first kPC's before and after renormalization.

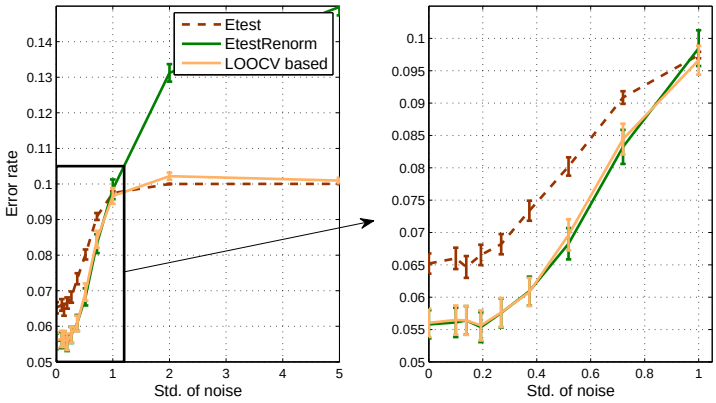


Figure 10: Mean error rates $\pm$ 1 standard deviation as a function of the noise level. The test error based on conventional kernel PCA projections, renormalized projections, and a LOOCV scheme is shown. Renormalization is seen to improve the performance, while LOOCV based renormalization prevents the classification error to blow up in the very low SNR regime.

5. Conclusion

Dimensionality reduction by PCA and kPCA can lack generalization due to training set variance inflation in the extremely ill-posed case when the sample size is much smaller than the input space dimension. In this work we have provided a simple geometric explanation for the main effect, namely that test points 'loose' their orthogonal projections, when their embedding is computed. This insight allowed for a speed-up of a previously proposed LOO scheme for renormalization. For kPCA we showed that the effects can be even more dramatic than in PCA, and we proposed a scheme for exact LOO renormalization of the embedding, and an approximate expression at lower cost. The viability of the new scheme was demonstrated for kPCA when used for dimensionality reduction both in simple synthetic data, in the USPS digit classification problem, and for fMRI brain state decoding.

Acknowledgments

We thank the reviewers of this manuscript and earlier versions for many useful comments. This research was supported by the Danish Lundbeckfonden through the Center for Integrated Molecular Brain Imaging (`www.cimbi.dk`).

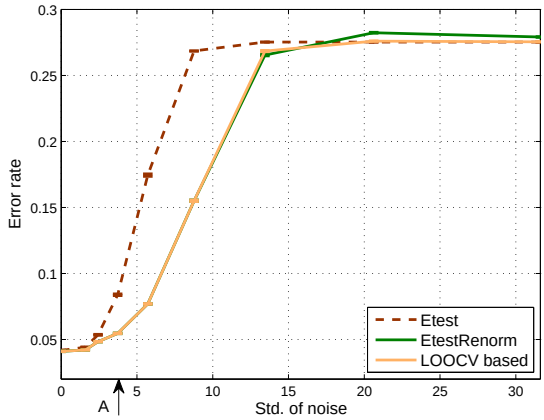


Figure 11: Mean error rates $\pm$ 1 standard deviation as a function of the noise level for fMRI data ($D = 16,384, N = 605$) . The test error based on conventional kernel PCA projections, renormalized projections, and a LOOCV scheme is shown. Renormalization is seen to clearly improve the performance. Arrow 'A' indicates the noise level used in Figure 12

Appendix A.

Let $\boldsymbol{u}_{N,k}$ be the k'th eigenvector of the covariance matrix on the full sample $\boldsymbol{\Sigma}_N$ and $\boldsymbol{u}_{N-1,k}$ be the corresponding eigenvector of LOO training set covariance matrix $\boldsymbol{\Sigma}_{N-1}$. In the following we use first order perturbation theory to show that

$$\boldsymbol{u}_{N-1,k}^T \cdot \boldsymbol{x}_N \approx \boldsymbol{u}_{N,k}^T \cdot \boldsymbol{x}_N^{\parallel} ,$$

where the data vector $\boldsymbol{x}$ has been split in its orthogonal and parallel components, $\boldsymbol{x}_N = \boldsymbol{x}_N^{\perp} + \boldsymbol{x}_N^{\parallel}$, relative to the subspace spanned by the training data. Thus, we are interested in the difference between $\boldsymbol{u}_{N,k}$ and $\boldsymbol{u}_{N-1,k}$. Simple manipulations of the covariance matrices lead to

$$\boldsymbol{\Sigma}_{N-1} = \boldsymbol{\Sigma}_N + \underbrace{\frac{1}{N-1}\boldsymbol{\Sigma}_N - \frac{1}{N}(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1})(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1})^T}_{O(\frac{1}{N})} .$$

By introducing the shorthand $\boldsymbol{A} = \boldsymbol{\Sigma}_{N-1}$ and $\boldsymbol{B} = \boldsymbol{\Sigma}_N$ we get

$$\boldsymbol{A} = \boldsymbol{B} + \delta \boldsymbol{C} , \tag{5}$$

where δ is of order $\frac{1}{N}$. Note that all matrices are symmetric. We now look at the k'th eigenvector of $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{B}\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k , \tag{6}$$

$$\boldsymbol{A}\boldsymbol{v}_k = \nu_k \boldsymbol{v}_k . \tag{7}$$

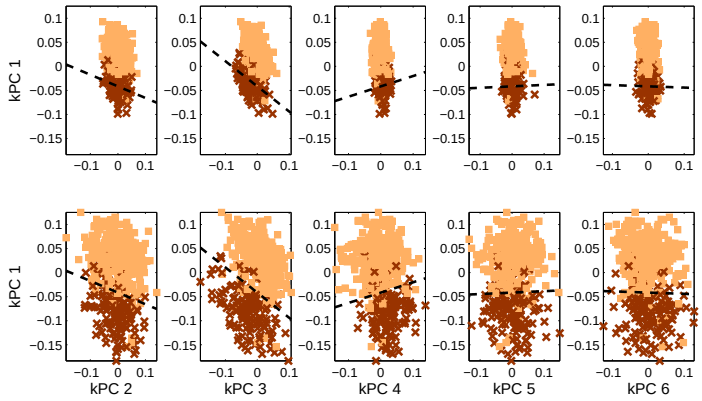


Figure 12: Test set projections of the fMRI data with Gaussian noise added as marked on Figure 11 ($\varepsilon_i = \mathcal{N}(0, 3.8^2)$). The top row shows the conventional projections, while the bottom row shows the projections after renormalization. The 'red class' indicates activation, while the blue observations are acquired during rest. The dashed line marks the linear discriminant. The scale is chosen as the 5th percentile of the mutual distances.

First order perturbation theory posits

$$\nu_k = \lambda_k + \delta\xi_k \,, \tag{8}$$

$$\boldsymbol{v}_k = \boldsymbol{u}_k + \delta\boldsymbol{w}_k \,. \tag{9}$$

That is, when going from N to $N-1$ samples we only have a small ($O(\frac{1}{N})$) change in eigenvalues and rotation of eigenvectors. Since all eigenvectors are orthonormal it follows that $\boldsymbol{u}_k \perp \boldsymbol{w}_k$, c.f.,

$$||\boldsymbol{v}_k||^2 = ||\boldsymbol{u}_k + \delta\boldsymbol{w}_k||^2 = \underbrace{||\boldsymbol{u}_k||^2}_{=1} + \underbrace{\delta^2}_{\approx 0}||\boldsymbol{w}_k||^2 + 2\delta\boldsymbol{u}_k^T\boldsymbol{w}_k = 1$$

$$\delta\boldsymbol{u}_k^T\boldsymbol{w}_k = 0 \,.$$

We now expand Equation (7) using Equation (5), (8) and (9)

$$\boldsymbol{A}\boldsymbol{v}_k = \nu_k\boldsymbol{v}_k \qquad \Rightarrow$$
$$(\boldsymbol{B} + \delta\boldsymbol{C})(\boldsymbol{u}_k + \delta\boldsymbol{w}_k) = (\lambda_k + \delta\xi_k)(\boldsymbol{u}_k + \delta\boldsymbol{w}_k) \,,$$

ignoring higher order terms of δ gives

$$\boldsymbol{B}\boldsymbol{u}_k + \delta\boldsymbol{C}\boldsymbol{u}_k + \delta\boldsymbol{B}\boldsymbol{w}_k = \lambda_k\boldsymbol{u}_k + \delta\lambda_k\boldsymbol{w}_k + \delta\xi_k\boldsymbol{u}_k \,,$$

Finally, exploiting Equation (6) reduces the above to

$$\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{B}\boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k + \xi_k \boldsymbol{u}_k \ . \tag{10}$$

We now look for an estimate of ξ_k by left multiplying with $\boldsymbol{u}_k^T$

$$\boldsymbol{u}_k^T \boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{B}\boldsymbol{w}_k = \lambda_k \boldsymbol{u}_k^T \boldsymbol{w}_k + \xi_k \boldsymbol{u}_k^T \boldsymbol{u}_k \ ,$$

using $||\boldsymbol{u}_k||^2 = 1$ and $\boldsymbol{u}_k \perp \boldsymbol{w}_k$ gives

$$\boldsymbol{u}_k^T \boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{B}\boldsymbol{w}_k = \xi_k \ ,$$

since $\boldsymbol{B}$ is symmetric, $\boldsymbol{u}_k$ is both a left and right singular vector. Hence, $\boldsymbol{u}_k^T \boldsymbol{B}\boldsymbol{w}_k = \lambda_k \boldsymbol{u}_k^T \boldsymbol{w}_k$=0. Thus finally, it follows that

$$\boldsymbol{u}_k^T \boldsymbol{C}\boldsymbol{u}_k = \xi_k \ . \tag{11}$$

Next, we find an estimate of $\boldsymbol{w}_k$ by left multiplying Equation (10) with $\boldsymbol{u}_j^T \ j \neq k$.

$$\boldsymbol{u}_j^T \boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{u}_j^T \boldsymbol{B}\boldsymbol{w}_k = \lambda_k \boldsymbol{u}_j^T \boldsymbol{w}_k + \xi_k \boldsymbol{u}_j^T \boldsymbol{u}_k \ ,$$

again we exploit the fact that $\boldsymbol{B}$ is symmetric and that $\boldsymbol{u}_j$ is orthogonal to $\boldsymbol{u}_k$, which gives

$$\boldsymbol{u}_j^T \boldsymbol{C}\boldsymbol{u}_k + \lambda_j \boldsymbol{u}_j^T \boldsymbol{w}_k = \lambda_k \boldsymbol{u}_j^T \boldsymbol{w}_k \ . \tag{12}$$

Assuming that span$\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_D\}$ = span$\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D\}$, that is, the $\boldsymbol{v}$-basis is a rotation of the $\boldsymbol{u}$-basis, which implies that $\boldsymbol{w}_k$ can be represented as a linear combination of the $\boldsymbol{u}$-vectors (or $\boldsymbol{v}$-vectors), leads to

$$\boldsymbol{w}_k = \sum_{m=1}^{D} h_{km} \boldsymbol{u}_m \ .$$

Due to orthonormality of the eigenvectors, we now realize that $h_{kk} = 0$ and $\boldsymbol{u}_j^T \boldsymbol{w}_k = \boldsymbol{u}_j^T \sum_{m=1}^{D} h_{km} \boldsymbol{u}_m$ will only be non-zero for $m = j$. Hence, Equation (12) reduces to

$$\begin{aligned} \boldsymbol{u}_j^T \boldsymbol{C}\boldsymbol{u}_k + \lambda_j h_{kj} = \lambda_k h_{kj} \qquad &\Rightarrow \\ h_{kj} &= \frac{\boldsymbol{u}_j^T \boldsymbol{C}\boldsymbol{u}_k}{\lambda_k - \lambda_j} \quad k \neq j \\ h_{kk} &= 0 \ . \end{aligned}$$

In the above we have assumed a nondegenerate system, that is, $\lambda_k \neq \lambda_j \ \forall k \neq j$. Thus, $\boldsymbol{w}_k$ can be expressed as

$$\boldsymbol{w}_k = \sum_{m=1 \neq k}^{N} \frac{\boldsymbol{u}_m^T \boldsymbol{C}\boldsymbol{u}_k}{\lambda_k - \lambda_m} \boldsymbol{u}_m \ , \tag{13}$$

where we used that $\boldsymbol{C}\boldsymbol{u}_k$ is only non-zero for $k \leq N$. We are now ready to return to Equation (8) and (9) inserting the expressions derived for ξ_k and $\boldsymbol{w}_k$ in Equation (11) and (13) respectively:

$$\nu_k = \lambda_k + \delta \boldsymbol{u}_k^T \boldsymbol{C} \boldsymbol{u}_k \tag{14}$$

$$\boldsymbol{v}_k = \boldsymbol{u}_k + \delta \sum_{m=1 \neq k}^{N} \frac{(\boldsymbol{u}_{\boldsymbol{m}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))(\boldsymbol{u}_{\boldsymbol{k}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))}{\lambda_k - \lambda_m} \boldsymbol{u}_m \, . \tag{15}$$

Equation (14) shows that the change in eigenvalue is indeed small ($O(\frac{1}{N})$) when going from N to $N-1$ samples. For the eigenvector perturbation, Equation (15), we can bound the squared length of the sum and obtain a similar result,

$$\begin{aligned} \left|\left| \frac{1}{N} \sum_{m=1 \neq k}^{N} \frac{(\boldsymbol{u}_{\boldsymbol{m}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))(\boldsymbol{u}_{\boldsymbol{k}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))}{\lambda_k - \lambda_m} \boldsymbol{u}_m \right|\right|^2 \leq \\ \frac{1}{N^2} ||\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}||^2 \left|\left| \sum_{m=1 \neq k}^{N} \frac{(\boldsymbol{u}_{\boldsymbol{m}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))}{\lambda_k - \lambda_m} \boldsymbol{u}_m \right|\right| = \\ \frac{1}{N^2} ||\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}||^2 \sum_{m=1 \neq k}^{N} \frac{|(\boldsymbol{u}_{\boldsymbol{m}}^T(\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}))|^2}{|\lambda_k - \lambda_m|^2} \leq \\ \frac{1}{N^2} \frac{2||\boldsymbol{x}_N - \boldsymbol{\mu}_{N-1}||^4}{|\Delta\lambda_k|^2} \, , \end{aligned}$$

where $\Delta\lambda_k$ is the spacing between the k'th eigenvalue and the closest neighbor, and the factor of two compensates for the missing k'th term in the sum, that is, the perturbation is of order $O(1/N)$

References

Michael Biehl and Andreas Mietzner. Statistical mechanics of unsupervised structure recognition. *Journal of Physics A-Mathematical and General*, 27(6):1885–1897, 1994.

Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007.

Mikio L. Braun, Joachim M. Buhmann, and Klaus-Robert Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9:1875–1908, 2008.

Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing*. 1977. ISBN 0-201-02596-5 (hardcover), 0-201-02597-3 (paperback).

David C. Hoyle and Magnus Rattray. A statistical mechanics analysis of gram matrix eigenvalue spectra. In *Lecture Notes in Computer Science, 17th Annual Conference on Learning Theory*, volume 3120, pages 579–593. Springer Verlag, 2004a.

David C. Hoyle and Magnus Rattray. Limiting form of the sample covariance eigenspectrum in pca and kernel pca. In *Advances in Neural Information Processing Systems 16*, pages 16–23. MIT Press, 2004b.

David C. Hoyle and Magnus Rattray. Principal-component-analysis eigenvalue spectra from data with symmetry-breaking structure. *Physical Review E*, 69(2):026124, 2004c.

David C. Hoyle and Magnus Rattray. Statistical mechanics of learning multiple orthogonal signals: Asymptotic theory and fluctuation effects. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 75(1):016101, 2007.

Jonathan J . Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

Robert Jenssen, Torbjörn Eltoft, Deniz Erdogmus, and Jose C. Principe. Some equivalences between kernel methods and information theoretic methods. *Journal of VLSI Signal Processing*, 45:49–65, 2006.

Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, 29(2):295–327, 2001.

Ulrik Kjems, Lars K. Hansen, and Stephen C. Strother. Generalizable singular value decomposition for ill-posed datasets. In *Advances in Neural Information Processing Systems 13*, pages 549–555. MIT Press, 2001.

Rudy Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3):233–246, 1989.

Sofia Mosci, Lorenzo Rosasco, and Alessandro Verri. Dimensionality reduction and generalization. In *Proceedings of the 24th International Conference on Machine Learning*, pages 657–664, 2007.

Peter Reimann, Chris Van den Broeck, and Geert J. Bex. A Gaussian scenario for unsupervised learning. *Journal of Physics A - Mathematical and General*, 29(13):3521–3535, 1996.

Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

John Shawe-Taylor and Christopher K. I. Williams. The stability of kernel principal components analysis and its relation to the process eigenspectrum. In *Advances in Neural Information Processing Systems 15*, pages 367–374. MIT Press, 2003.

Jack W. Silverstein and Patrick L. Combettes. Signal-detection via spectral theory of large dimensional random matrices. *IEEE Transactions on Signal Processing*, 40(8):2100–2105, 1992.

Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Advances in Neural Information Processing Systems 18*, pages 1649–1656. MIT Press, 2006.

Appendix F

Restoring the Generalizability of SVM Based Decoding in High Dimensional Neuroimage Data

Trine Julie Abrahamsen and Lars Kai Hansen. Restoring the Generalizability of SVM Based Decoding in High Dimensional Neuroimage Data. In Georg Langs, Irina Rish, Moritz Grosse-Wentrup, and Brian Murphy, editors, *Machine Learning and Interpretation in Neuroimaging*, Lecture Notes in Computer Science, pages 256–263. Springer Berlin Heidelberg, 2012.
doi:10.1007/978-3-642-34713-9_32.

Restoring the Generalizability of SVM Based Decoding in High Dimensional Neuroimage Data

Trine Julie Abrahamsen and Lars Kai Hansen

Technical University of Denmark, Richard Petersens Plads, 2800 Lyngby, Denmark
{tjab,lkh}@imm.dtu.dk

Abstract. Variance inflation is caused by a mismatch between linear projections of test and training data when projections are estimated on training sets smaller than the dimensionality of the feature space. We demonstrate that variance inflation can lead to an increased neuroimage decoding error rate for Support Vector Machines. However, good generalization may be recovered in part by a simple renormalization procedure. We show that with proper renormalization, cross-validation based parameter optimization leads to the acceptance of more non-linearity in neuroimage classifiers than would have been obtained without renormalization.

Keywords: Support Vector Machines, generalizability, variance inflation, imbalanced data.

1 Introduction

The Support Vector Machine (SVM) is, by a margin, the most popular classifier in neuroimaging. The SVM has advantages over other classifiers in terms of computation, performance, and it typically involves only few control parameters. Furthermore, the SVM is rooted in learning theory, which has led to magic thinking like *'Moreover, unlike other machine learning methods, SVMs generalization error is related not to the input dimensionality of the problem, but to the margin with which it separates the data'* [9]. However, many neuroimage decoding problems are highly ill-posed involving small samples in high dimensions [10] and machine learning is only beginning to understand how the curse of dimensionality challenges the SVM [6].

Variance inflation is one of the challenges that can appear in high dimensions. In [1] we show how kernel Principal Component Analysis (kPCA) may suffer from variance inflation in high dimensional data. Variance inflation is caused by a mismatch between linear projections of test and training data when learning is still incomplete due to limited sample size. In the following we will demonstrate that the variance inflation problem extends to SVMs and we will adapt the framework developed for kPCA to restore the proper variation and hence generalizability of SVMs. We experimentally validate our approach on both the benchmark USPS handwritten digit classification problem [7] and on functional Magnetic Resonance Imaging (fMRI) data.

G. Langs et al. (Eds.): MLINI 2011, LNAI 7263, pp. 256–263, 2012.

While variance inflation is expected in ill-posed data in general, it may be further amplified in imbalanced data, i.e., classification problems with different probabilities of the classes [1]. The problems related to imbalanced data are well known, see [17] for a review, however, the underlying mechanism relating to variance inflation has not been noted earlier. The most common strategies to reduce the advantage of the majority class for imbalance data include various schemes for oversampling the minority class, undersampling the majority class or a combination of the two (e.g. [3,5,14,16]). Another approach is to apply instance weighting (or cost-sensitive learning), where the two classes are assigned different weights in the SVM [2,12]. Other approaches include combining oversampling and different error costs [2], while [18] adjusted the classification boundary toward the majority class by modifying the kernel matrix, however the time complexity does not scale well to very large data sets [16]. Recently [11] suggested a kernel transformation to unevenly enlarge the feature space in proximity of the boundary region to compensate for the class skewness.

The rest of this paper is organized as follows. Section 2 presents the renormalization scheme for restoring the generalizability of the SVM, while Section 3 describes the experimental results. Finally, Section 4 concludes the paper.

2 Restoring Generalizability

Due to variance inflation, the output of the SVM prior to thresholding, here referred to as the decision value,

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i \in \mathcal{S}} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{1}$$

of the test data does not follow the same distribution as for the training data. In Equation (1) α_i and y_i are the Lagrange parameters and labels respectively, for the i'th support vector, while $k(.,.)$ is the kernel function and b is the threshold parameter.

This training set test set mismatch may potentially lead to a increased misclassification rate, as illustrated in Figure 1 and in the right panel of figure 2.

We propose a non-parametric procedure for restoring the variation in the decision values of the test set, so that a classifier adapted to the inflated training data will perform well in test. The problem of variance inflation and the proposed procedure is illustrated in Figure 1. Basically, we assume that there exist a monotonic transformation between the decision values of the training and test set. The problem of calibrating for an unknown monotonic gray scale transformation is a common operation in image processing. Equalizing two equal sized samples, simply involves sorting both and assigning the sorted test decision values the sorted training decision values. The transformation can formalized as follows. Let $H(f)$ be the cumulative distribution of the decision values, f, in the training data, and let the decision values for test samples take values $g(n)$,

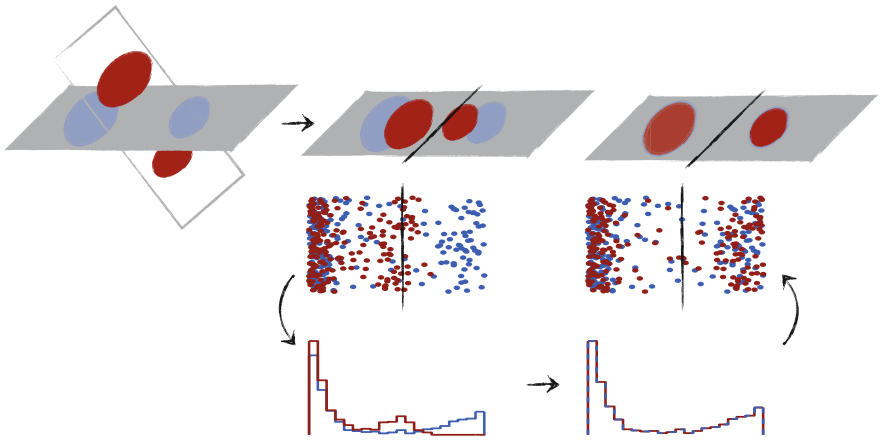


Fig. 1. Illustration of the variance inflation problem and the proposed non-parametric procedure for restoring generalizability. The blue cluster constitute the training set, while the blue clusters are the test data. The left panel illustrates the possible mismatch between projections of test and training data when the training data set is smaller than the dimension of the input data space. The middle panel shows how this mismatch causes the decision values to follow different probability laws in the training and test sets. Finally, the right panel illustrates the effect of restoring the variation of the test set by the histogram equalization based procedure

$n = 1, ..., N$. Let $I(n)$ be the index of sample n in a sorted list of the test set values. Then the renormalized value of the decision value of the test datum n, is

$$\widetilde{g(n)} = H^{-1}(I(n)/N).$$

In the simplest case of equally sized training and test sets, the renormalized decision values are obtained by the simple relation:

$$\widetilde{g(n)} = f_{sort}(I(n)), \tag{2}$$

where f_{sort} is the sorted list of the decision values of the training data. The renormalization procedure only requires two additional $\mathcal{O}(N \log(N))$[1] operations for sorting the decision values of the training and test sets. In cases where the number of observations in the training and test sets differ, we interpolate the renormalization function [1]. The algorithm for non-parametric renormalization is summarized in Algorithm 1.

Please note that the transformation relates to the distribution of the decision values, hence we do not consider nor estimate approximate posterior probabilities to implement the renormalization procedure, c.f., [13].

[1] This is the average time complexity of quicksort.

Algorithm 1. Renormalization in SVM

Compute the decision values for training and test set: {we use LIBSVM [4]}
for $n = 1$ to N_{tr} **do**
  $\boldsymbol{f}_{tr}^{n} \leftarrow \sum_{i \in \mathcal{S}} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}_n) + b$ {where $\mathcal{S}$ is the set of SVs}
end for
for $n = 1$ to N_{te} **do**
  $\boldsymbol{f}_{te}^{n} \leftarrow \sum_{i \in \mathcal{S}} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}_n) + b$
end for
$[\boldsymbol{f}_{sort}, \ ] \leftarrow \text{sort}(\boldsymbol{f}_{tr})$ {ascending order}
$[\ , I] \leftarrow \text{sort}(\boldsymbol{f}_{te})$ {ascending order}
if $N_{tr} = N_{te}$ **then**
  $\boldsymbol{h} \leftarrow \boldsymbol{f}_{sort}$
else $\{N_{tr} \neq N_{te}\}$
  $\boldsymbol{h} \leftarrow \text{spline}\big([1 : N_{tr}], \boldsymbol{f}_{sort}, \text{linspace}(1, N_{tr}, N_{te})\big)$ {interpolate to create N_{te} values of $\boldsymbol{f}_{sort}$ in the interval $[1 : N_{tr}]$}
end if
for $n = 1$ to N_{te} **do**
  $\tilde{\boldsymbol{g}}_{te}^{I(n)} \leftarrow \boldsymbol{h}^n$ {renormalized decision values of the test set, see Equation (2)}
end for

3 Experiments

We report briefly on two data sets, one benchmark and one functional neuroimage data set. We used the LIBSVM [4] software and the Gaussian kernel, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$, where γ is the scale parameter controlling the non-linearity of the kernel map, for all experiments. The performance was measured using standard accuracy as well as the G-mean metric [8]. The G-mean is the geometric mean of the sensitivity and specificity (G-mean $= \sqrt{\text{sensitivity} \cdot \text{specificity}}$), and is often used on highly imbalanced data sets as the accuracy measure degenerates with increasing imbalance.

USPS Handwritten Digit Data

As a classic benchmark data set in supervised kernel methods we used the USPS data consisting of 16×16 pixels handwritten digits[2]. For each digit we randomly chose 10 examples for training and another 10 examples for testing. The soft margin slack parameter, C, was set to 1 or 10, while the scale parameter of the kernel was chosen so $1/\sqrt{\gamma}$ was the $5th$ percentile of the mutual distances of the data points. We added Gaussian noise, $\mathcal{N}(0, \sigma^2)$, with $\sigma \in [0, 1]$, and for every noise level, 200 random training and test sets where drawn. We classify digit 8 versus the rest, leading to an imbalance ratio of 1:9. Z-score normalization was used prior to classification to transform the features to a distribution with zero mean and unitary variance by: $\boldsymbol{x}_{zs} = (\boldsymbol{x} - \text{mean}(\boldsymbol{x}))/\text{std}(\boldsymbol{x})$.

[2] The USPS data set is described in [7] and can be downloaded from `www.kernel-machines.org`

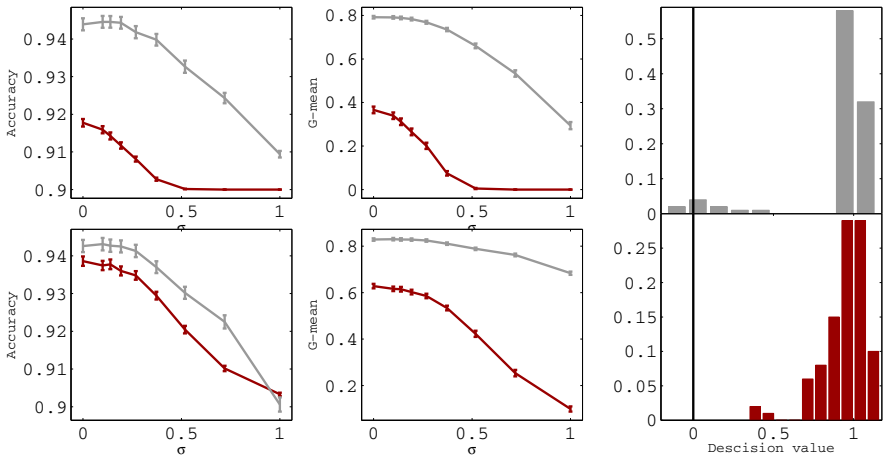


Fig. 2. Mean performance measures ±1 std as a function of the noise level for the USPS data. The test accuracy is shown in red while the renormalized test accuracy is shown in gray. The left and middle panels show the accuracy and the G-mean respectively. In the top panel the slack parameter, C, is set to 1, while $C = 10$ in the lower panels. The right panel shows an example of the histogram before and after renormalization (for $C = 1$ and a noise level of $\sigma = 0.27$). it is evident how renormalization leeds to improved performance for all noise levels.

The results are summarized in Figure 2, where we show both the accuracy and G-mean before and after renormalization in the two left panels. It is evident that renormalization improves the classification for all noise levels and provides meaningful classification even when the unnormalized predictions are all classified as the majority class. Furthermore, it can be seen how the renormalized predictions are less affected by the choice of slack parameter. The right panel in Figure 2 shows an example of the histogram of the decision values before and after the renormalization procedure. Both the variance inflation of the original SVM outputs and the positive effect of restoring the test set variance are evident.

Visual Paradigm fMRI Data

The participants were subjected to four visual conditions presented on a monitor: no visual stimulation (NO), reversing checkerboard on the left half of the screen (LEFT), reversing checkerboard on the right half of the screen (RIGHT), and reversing checkerboard on both halves of the screen (BOTH). The data set consists of 12 scans from each state for 6 subjects, giving a total of 288 observations. The data was masked with a rough whole-brain mask leading to D=75,257 voxels. The data acquisition and preprocessing pipeline is described in detail in [15]. We construct an imbalanced problem by to different schemes. First, we

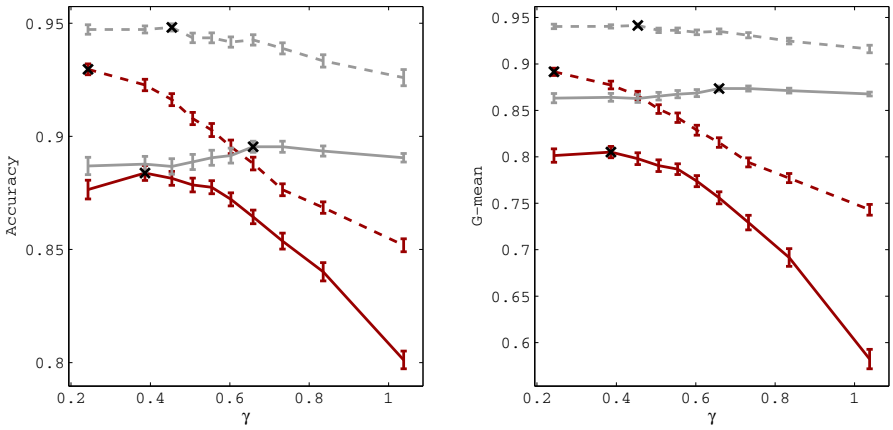


Fig. 3. Mean performance measures ± 1 std as a function of kernel hyperparameter for the fMRI data. Higher values of γ lead to more non-linear kernel embeddings. The left and right panel shows the accuracy and the G-mean respectively. The dashed lines correspond to the scheme where data with no stimuli are omitted, while the full lines show the performance on the subsampled data. The test accuracy is shown in red while the renormalized test accuracy is shown in gray. The black crosses indicate the optimal kernel hyperparameter. Renormalization is seen to improve performance and notably it leads to more non-linear optimal kernels as the optimal scale parameters chosen by cross-validation are increased.

exclude all scans with no stimuli and classify (RIGHT or LEFT) versus BOTH leading to a ratio of 1:2. Secondly, we classify (RIGHT or LEFT) versus (NO or BOTH), where imbalance is achieved by excluding part of the RIGHT/LEFT observations by random subsampling. We use an imbalance ratio of 1:2.5. In both schemes we fix $C = 10$ (the slack parameter was fund to have little to no effect on performance for sufficient large C values), while the scale of the kernel is varied from the $95th$ to $5th$ percentile of the mutual distances ($\gamma \in [0.24, 1.05]$). For every value of γ, we split the data in 3 subjects for training and 3 subjects for testing and report the accuracy and G-mean measure on the test set before and after renormalization.

The results are summarized in Figure 3. The improved performance after renormalization is significant for both schemes. Furthermore, it should be noted how renormalization leads to a more non-linear optimal kernel (larger optimal γ) determined in cross-validation. As the scale parameter, γ, goes to zero, the Gaussian kernel approaches the simple linear kernel. Hence, the results shown for small γ values in Figure 3 indicate that even for linear kernels applying the renormalization scheme will lead to improved performance.

4 Conclusion

In ill-posed imbalanced data, variance inflation of the training set may cause seriously degraded generalizability in SVM decoding. We proposed a non-parametric scheme for renormalization of the decision values and thereby restoring generalizability. Viability was demonstrated on the USPS handwritten digits and in fMRI brain state decoding. In both examples we showed how the renormalization procedure improved the performance. Furthermore, renormalization suggested a more non-linear optimal kernel embedding in the brain state decoding experiment.

Acknowledgments. The fMRI data set was acquired and preprocessed at the Danish Research Centre for Magnetic Resonance, Hvidovre Hospital, Denmark. The Simon Spies Foundation is acknowledged for donation of the Siemens Trio scanner. We thank the authors of [15] for sharing the data set.

References

1. Abrahamsen, T.J., Hansen, L.K.: A cure for variance inflation in high dimensional kernel principal component analysis. Journal of Machine Learning Research, 2027–2044 (July 2011)
2. Akbani, R., Kwek, S., Japkowicz, N.: Applying Support Vector Machines to Imbalanced Datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
3. Castro, C.L., Carvalho, M.A., Braga, A.P.: An Improved Algorithm for SVMs Classification of Imbalanced Data Sets. In: Palmer-Brown, D., Draganova, C., Pimenidis, E., Mouratidis, H. (eds.) EANN 2009. CCIS, vol. 43, pp. 108–118. Springer, Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-03969-0_11`
4. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
6. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer Series in Statistics. Springer (2009), corr. 3rd printing 5th printing, edn.
7. Hull, J.J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(5), 550–554 (1994)
8. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Proc. 14th International Conference on Machine Learning, pp. 179–186. Morgan Kaufmann (1997)
9. Kwok, J.T.-Y.: Moderating the outputs of support vector machine classifiers. IEEE Transactions on Neural Networks 10, 1018–1031 (1999)
10. Lautrup, B., Hansen, L.K., Law, I., Svarer, C., Strother, S.C.: Massive weight sharing: A cure for extremely ill-posed problems. In: Supercomputing in Brain Research: From Tomography to Neural Networks. World Scientific Pub. Corp., pp. 137–148. World Scientific (1994)

11. Maratea, A., Petrosino, A.: Asymmetric Kernel Scaling for Imbalanced Data Classification. In: Fanelli, A.M., Pedrycz, W., Petrosino, A. (eds.) WILF 2011. LNCS, vol. 6857, pp. 196–203. Springer, Heidelberg (2011)
12. Osuna, E.E., Freund, R., Girosi, F.: Support vector machines: Training and applications. Tech. rep., MIT, Cambridge (1997)
13. Platt, J.C.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Schlkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods Support Vector Learning, pp. 61–74 (2000)
14. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for SVMs: a case study. SIGKDD Explor. Newsl. 6(1), 60–69 (2004)
15. Rasmussen, P.M., Madsen, K.H., Lund, T.E., Hansen, L.K.: Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. NeuroImage 55(3), 1120–1121 (2011)
16. Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S.: SVMs Modeling for Highly Imbalanced Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39(1), 281–288 (2009)
17. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explor. Newsl. 6, 7–19 (2004)
18. Wu, G., Chang, E.Y.: KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution. IEEE Transactions on Knowledge and Data Engineering 17(6), 786–795 (2005)

Appendix G

Variance Inflation in High Dimensional Support Vector Machines

Trine Julie Abrahamsen and Lars Kai Hansen. Variance Inflation in High Dimensional Support Vector Machines. *Pattern Recognition Letters*, Accepted, 2013.

Variance Inflation in High Dimensional Support Vector Machines

Trine Julie Abrahamsen, Lars Kai Hansen

DTU Compute, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Abstract

Many important machine learning models, supervised and unsupervised, are based on simple Euclidean distance or orthogonal projection in a high dimensional feature space. When estimating such models from small training sets we face the problem that the span of the training data set input vectors is not the full input space. Hence, when applying the model to future data the model is effectively blind to the missed orthogonal subspace. This can lead to an inflated variance of hidden variables estimated in the training set and when the model is applied to test data we may find that the hidden variables follow a different probability law with less variance. While the problem and basic means to reconstruct and deflate are well understood in unsupervised learning, the case of supervised learning is less well understood. We here investigate the effect of variance inflation in supervised learning including the case of Support Vector Machines (SVM) and we propose a non-parametric scheme to restore proper generalizability. We illustrate the algorithm and its ability to restore performance on a wide range of benchmark data sets.

Keywords: Variance inflation, SVM, generalization

1. Introduction

Many machine learning models, supervised and unsupervised, are based on the Euclidean distance to or orthogonal projection on a training set of feature vectors. If the dimension of the vector space D is high and the training set size N small ($N \ll D$) we face the problem that the training

Email address: tjab@imm.dtu.dk, +45 45253888 (Trine Julie Abrahamsen)

feature vectors do not span the whole of feature space. Hence, when applying the estimated model to future test data it is effectively blind to the subspace orthogonal to the space spanned by the training data. This implies that the variance of the projections in the training set will be much larger than the variance in test data, and more generally that hidden variables will tend to follow different probability laws in training and test data. While the problem and means to reconstruct and calibrate are quite well understood in unsupervised learning (Kjems et al., 2001; Abrahamsen and Hansen, 2011), the similar problem in supervised learning has only recently been noted for decoding of fMRI data (Abrahamsen and Hansen, 2012). In this paper we provide a more detailed analysis of variance inflation in supervised learning. We expand both on the theoretical analysis, in particular we provide an exact expression for the variance inflation in a simple linear regression model, we provide a calibration mechanism based on leave-one-out that can work also if we do not have access to a test data set, and finally, we demonstrate variance inflation and the effects of our calibration procedures in 18 benchmark data set with a range of N/D ratios.

2. Variance inflation in simple linear models

First, to get a more detailed insight into how variance inflation results from the 'lost projection' present ill-posed learning $N < D$, we investigate simple linear regression.

Let a linear regression problem be defined as follows, $y = \mathbf{w}^\top \mathbf{x} + \epsilon = \sum_{d=1}^{D} w_d x_d + \epsilon$, based on D real input variables collected in the vector $\mathbf{x}$. We will assume that the elements of $\mathbf{x}$ are zero mean, unit variance, i.i.d., normal variates, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, and that the noise is distributed as $\epsilon \sim \mathcal{N}(0, \sigma^2)$, and independent of the input variables. Estimation of the regression coefficients $\mathbf{w}$ is based on N $(y, \mathbf{x})$ pairs, sampled independently from the model with a fixed set of true parameters $\mathbf{w}_0$ The least squares estimator

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^{N} \left(y_n - \mathbf{w}^\top \mathbf{x}_n \right)^2, \tag{1}$$

is unique in the well-posed case. For the ill-posed situation, with $N < D$, we use the pseudo-inverse solution, i.e., the unique solution constrained to the linear subspace spanned by the N input vectors (Penrose and Todd, 1956).

The expected generalization error,

$$G(N) = E_{y,\mathbf{x}}\left\{E_N\left\{\left(y - \mathbf{w}_N^\top \mathbf{x}\right)^2\right\}\right\}, \tag{2}$$

where $E_{y,\mathbf{x}}$ refers to the expectation with respect to a new test data point, and E_N refers to the expectation with respect to the N training points, of the least mean square estimator is known exactly for all D, N (Hansen, 1993; Barber et al., 1995),

$$G(N) = \begin{cases} \left(1 - \frac{N}{D}\right)||\mathbf{w}_0||^2 + \frac{D-1}{D-N-1}\sigma^2 & N < D-1, \\ \infty & D-1 \leq N \leq D+1, \\ \frac{N-1}{N-D-1}\sigma^2 & N > D+1. \end{cases}$$

Note that the 'excess' error, $\Delta G = G - \sigma^2$, for small values of N/D, is dominated by the 'lost projection' bias term $||\mathbf{w}_0||^2 - E_N\left\{||\widehat{\mathbf{w}}||^2\right\} = \left(1 - \frac{N}{D}\right)||\mathbf{w}_0||^2$.

Here we are interested in analyzing the training set inflation of projections and focus on the predictions made by the pseudo-inverse solution, $\widehat{\mathbf{w}} = \sum_{m,n=1}^{N} \mathbf{x}_n (K^{-1})_{n,m} y_m$, were we have introduced the $N \times N$ kernel matrix $K_{m,n} = \mathbf{x}_m^\top \mathbf{x}_n$. Note the kernel matrix is non-singular with probability one for $D \geq N$. By assuming that the input data is zero mean ($\sum_n \mathbf{x}_n = 0$), and noting that $\widehat{\mathbf{w}}^\top \mathbf{x}_n = y_n$ in the underdetermined case, when $\hat{\mathbf{w}}$ is given by the pseudo-inverse, the variance of the training projections is given by

$$\sigma^2\left(\widehat{\mathbf{w}}^\top \mathbf{x}_n\right) = 1/N \sum_{n=1}^{N} y_n^2, \tag{3}$$

with expectation, $E_N\left\{1/N \sum_{n=1}^{N} y_n^2\right\} = ||\mathbf{w}_0||^2 + \sigma^2$, while the variance of test projections is given by

$$E_{\mathbf{x}}\left\{E_N\left\{\widehat{\mathbf{w}}^\top \mathbf{x}\right\}\right\} = E_N\left\{||\widehat{\mathbf{w}}||^2\right\} = \frac{N}{D}||\mathbf{w}_0||^2. \tag{4}$$

Hence, the variance of the predicted output is reduced by more than N/D relative to the training output.

In (Abrahamsen and Hansen, 2011) it was indicated that the variance inflation problem of an unsupervised pre-processing method as PCA may seriously affect a subsequent classifier. For a classifier trained directly on the high-dimensional input, e.g., Fisher's linear discriminant (FLD), this has

similar implications. Note, the FLD may be obtained by means of a linear regression trained by least mean squares, if we carefully tune the target values to reflect possible imbalance in class occurrence (Christopher, 1996). In this case the model is augmented with a bias weight b, and binary decisions are obtained as $y(\mathbf{x}) = \text{sign}\left(\widehat{\mathbf{w}}^\top \mathbf{x} + b\right)$. Based on the analysis of the pseudo-inverse solution we conclude that the decision values $f \equiv \widehat{\mathbf{w}}^\top \mathbf{x} + b$, i.e., the arguments of the 'sign' function, are distributed differently in training and test sets potentially leading to excess classification errors.

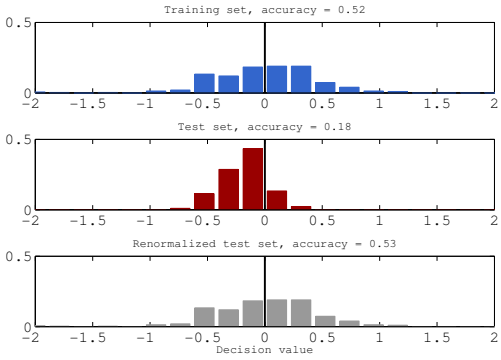


Figure 1: Illustration of the variance inflation phenomena in simulated data. The plots show the distribution of the decision values, $\boldsymbol{f}$, of a FLD. The top panel is the training data, the middle panel is the test data, and the lower panel shows the result after applying the non-parametric scheme for restoring the variation as described in following section. The inflated variance of the training data compared to the test data is evident.

The latter effect is demonstrated in a simulated data set in Figure 1. An imbalanced linearly separable data set was created with noisy labels $y = \text{sign}(\mathbf{w}_0^\top \mathbf{x} + b + \epsilon)$, with $\epsilon \sim \mathcal{N}(0, 10^{-4})$, $D = 2000, N = 300$, and all elements of $\mathbf{w}_0 = 1/\sqrt{D}$. The inflated variance of the training data in the top panel is evident when comparing to the test set in the middle panel. The lower panel shows the result of the non-parametric renormalization scheme described below, which effectively restores the variance and improves the test classification accuracy.

3. Generalizability in SVMs

While Support Vector Machines (SVM) (Vapnik, 1995) are indeed very popular for classification, we are only starting to understand how the curse of dimensionality affects SVM performance (Hastie et al., 2009).

In order to understand the variance inflation problem in relation to SVMs we briefly recall the settings for this non-linear classification method. Let $\mathcal{H}$ be the Reproducing Kernel Hilbert Space (RKHS) induced by a kernel function, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j)$, where $\varphi : \mathcal{X} \to \mathcal{H}$ is possibly non-linear map from input space to the high dimensional RKHS.

Given a set of training data $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, $\boldsymbol{x} \in \mathbb{R}^D$, $y \in \{1, -1\}$, the SVM problem can be expressed as[1]

$$\min_{\boldsymbol{w} \in \mathcal{H}} \sum_{i=1}^{N} \max(0, 1 - y_i \boldsymbol{w}^\top \varphi(\boldsymbol{x}_i)) + \lambda \|\boldsymbol{w}\|^2, \tag{5}$$

where the first term corresponds to the Hinge-loss function. By applying the representer theorem (Kimeldorf and Wahba, 1971; Schölkopf and Smola, 2001), $\boldsymbol{w}$ can be rewritten as $\boldsymbol{w} = \sum_{i=1}^N \alpha_i \varphi(\boldsymbol{x}_i)$, and the kernel trick is made applicable. The subset of training samples for which $\alpha \neq 0$ defines the margin, and are referred to as support vectors (SV). The set of SVs will be denoted $\mathcal{S}$. Note that the slack term controlling the width of the margin is given by $C = \frac{1}{\lambda}$. The test phase of the SVM consists of finding

$$\operatorname{sign}\big(y(\boldsymbol{x})\big) = \operatorname{sign}\Big(\sum_{i \in \mathcal{S}} y_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b\Big), \tag{6}$$

where α_i and y_i are the Lagrange parameters and labels for the i'th support vector, respectively. We focus on the Gaussian kernel of the form $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$, where γ is the scale parameter controlling the non-linearity of the kernel embedding. The Gaussian kernel is by a margin the most popular non-linear kernel because of its generality. We defer the investigation of the interesting case of other specialized kernels to a later report. We will in the following refer to $y(\boldsymbol{x})$ as the decision value of a given observation, $\boldsymbol{x}$.

Variance inflation is caused by a mismatch between the subspaces spanned by the training and test data. Ie., when learning is still incomplete due to

[1] The bias term, b, has been omitted for notational convenience.

a limited samples size relative to the problem complexity. While ill-posed problems generally suffer from variance inflation, we expect generalizability to deteriorate significantly when the data have different prior class probabilities.

The fundamental challenges faced with imbalanced problems are well known, see, e.g., (Weiss, 2004) for a review, and, several schemes for adjusting SVMs to generalize well on imbalanced data have been suggested. The most common strategies include oversampling the minority class, undersampling the majority class or a combination of the two (e.g., (Castro et al., 2009; Chawla et al., 2002; Raskutti and Kowalczyk, 2004; Tang et al., 2009)). However, the underlying mechanism and effects of variance inflation is yet to be recognized.

In the following we claim that the variance inflation problem extends from kernel PCA to SVMs, and in line with (Abrahamsen and Hansen, 2011), we propose a framework for restoring the test set variation and hence the generalizability of the SVM. In order to experimentally validate our approach, we report performance on 18 real life data sets from the University of California, Irvine Machine Learning Repository (Frank and Asuncion, 2010) and the Kent Ridge Biomedical Data Set Repository[2].

Variance inflation causes the decision values of the test observations to follow a distribution different than for the training data as we saw in the analysis of the linear models. This mismatch between training and test set distributions may potentially lead to increased misclassification rate, as illustrated in Figure 2.

In order to restore the variation so a classifier trained on the inflated training data performs well on test data, we turn to histogram specification. The problem of calibrating for an unknown monotone transformation is a common operation in image processing, and is used, for example, to enhance the contrast of an image by standardizing the pixel histogram (Gonzalez and Wintz, 1977) as was proposed in (Abrahamsen and Hansen, 2011) for unsupervised learning. Equalizing two equally sized samples, simply involves sorting both and assigning the sorted test set decision values the sorted values of the training decision values. This procedure is easily seen to force the test data to follow the same distribution as the training data, without changing the level sets (relative ordering) of the test data.

[2]Available at `http://datam.i2r.a-star.edu.sg/datasets/krbd/`

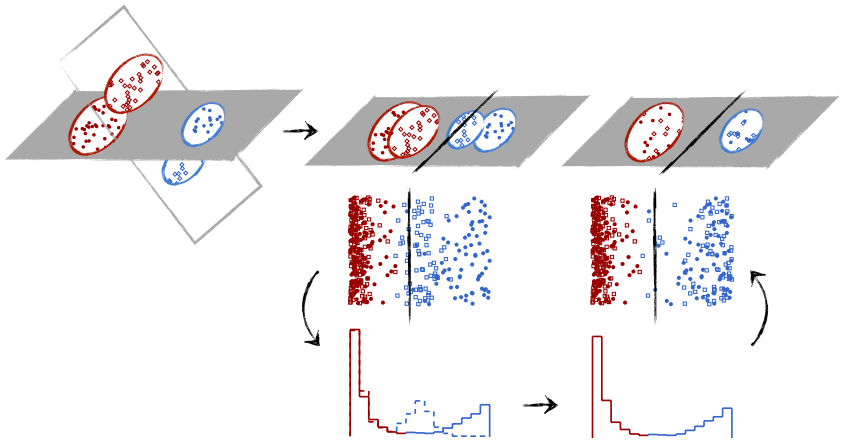


Figure 2: Illustration of the variance inflation problem and the proposed non-parametric procedure for restoring generalizability. The two classes are marked in red and blue respectively, while dots constitute the training set and squares indicate the test data. The left panel illustrates the possible mismatch between the test and training data subspaces when learning is incomplete. The middle panel shows how this mismatch causes the decision values to follow different probability laws in the training and test sets when projected onto the training set subspace. Finally, the right panel illustrates the effect of restoring the variation of the test set by the histogram specification based procedure. (Abrahamsen and Hansen, 2012)

In the following we show how the subspace mismatch lead to a common scaling factor due to the lost orthogonal projection for all test observations. We propose a straightforward framework for restoring the variation in the test set decision values and hence the generalizability of the SVM as illustrated in Figure 2. Finally, we provide a means for how the transformation can be learned in a cross-validation setting, and propose a faster approximation applicable when the training and test sets have equal prior distributions.

3.1. LOO procedure to learn renormalization function

We assume that there exists a monotonic transformation between the test set and the training set decision values that restores the variation of the test set. We propose a leave-one-out (LOO) procedure for learning this transformation.

Initially, we compute N test decision values by repeatedly training a SVM on the $N-1$ sub training sets. The LOO step for the n'th training sample, $\boldsymbol{x}_n$, consists of training a SVM on the $N-1$ other training samples and then determining the decision value, $y(\boldsymbol{x}_n) = f_n$, of $\boldsymbol{x}_n$. We now assume that there exists a monotonic transformation between the N training decision values, $\{d_1, \ldots, d_N\}$ obtained when training on the full training sample and the corresponding N LOO test decision values, $\{f_1, \ldots, f_N\}$. Thus our renormalization task now consists of learning a function, $h : \mathbb{R} \to \mathbb{R}$, that maps the LOO test decision values onto the training decision values. Now, let $\boldsymbol{d}_s \in \mathbb{R}^N$ be the sorted values of $\{d_1, \ldots, d_N\}$ and $I(i)$ the index of sample i in a sorted list of the LOO test set decision values, $\{f_1, \ldots, f_N\}$. Then $h(f_i) \equiv d_s(I(i))$. We learn the transfer function, $\hat{h}$ by fitting an increasing spline function with 10 equidistant knots.

Since the learning phase involves training N separate SVM's, it is desirable to reduce the complexity. In the following we suggest to include a common scaling factor or apply histogram specification on the decision values to reduce the computational burden.

3.2. Orthogonal projection factor

As a simple approximation to the LOO procedure, we consider adjusting for the lost projection contribution by a common scaling factor. In $N \ll D$ problems, we expect the lack of generalizability to be partially explained by such a factor.

Define the orthogonal and parallel components of the j'th test observation as, $\boldsymbol{x}_j = \boldsymbol{x}_j^{\parallel} + \boldsymbol{x}_j^{\perp}$ relative to the subspace of $\mathcal{X}$ spanned by the training data. Considering the squared distance in the Gaussian kernel and splitting the test observation in the orthogonal and parallel components in Equation (6) leads to

$$y(\boldsymbol{x}_j) = \sum_{i \in \mathcal{S}} y_i \alpha_i \exp\Big(-\gamma(||\boldsymbol{x}_i - \boldsymbol{x}_j^{\parallel}||^2 + ||\boldsymbol{x}_j^{\perp}||^2)\Big) + b \tag{7}$$

$$= \Big(\exp\Big(-\gamma||\boldsymbol{x}_j^{\perp}||^2\Big) \sum_{i \in \mathcal{S}} y_i \alpha_i \exp\Big(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j^{\parallel}||^2\Big)\Big) + b. \tag{8}$$

It is clear, how the test values all require a common factor of $\exp(-\gamma||\boldsymbol{x}_j^{\perp}||^2)$, which can be arbitrary small in the non-linear regime (large γ). Hence, we suggest to apply a correction factor in order to account for the lost orthogonal projection leading to the following expression for the adjusted decision value

$$y^f(\boldsymbol{x}_j) = \exp(\gamma||\boldsymbol{x}_j^{\perp}||^2)\Big(y(\boldsymbol{x}_j) - b\Big) + b = \left(\exp\left(\gamma||\boldsymbol{x}_j^{\perp}||^2\right)\sum_{i\in\mathcal{S}} y_i\alpha_i k_{ij}\right) + b. \tag{9}$$

3.3. Histogram specification

When the training and test sets have the same prior distribution we can use standard histogram specification to restore the variation in the test set. The problem of calibrating for an unknown monotonic transformation is a common task in image processing equivalent of equalizing two equal sized images. Let $H(\boldsymbol{d})$ be the cumulative distribution of the training set decision values, $\{d_1, \ldots, d_N\}$ and let the decision values of a test sample drawn from the same prior distribution take values $\{g_1, \ldots, g_N\}$. Now let $I(j)$ be the index of sample j in a sorted list of the test set values. Then the renormalized decision value, $y^h(\boldsymbol{x}_j)$, of test observation j is given by

$$y^h(\boldsymbol{x}_j) = H^{-1}(I(j)/N).$$

Hence, in the case of equal sized training and test sets, renormalization of the entire test set is achieved by

$$y^h(\boldsymbol{x}_j) = d_s(I(j)), \tag{10}$$

where $\boldsymbol{d}_s \in \mathbb{R}^N$ is a sorted list of the training set decision values, $\{d_1, \ldots, d_N\}$ as previously defined. The algorithm for non-parametric renormalization is summarized in Algorithm 1. This renormalization procedure only requires two additional $\mathcal{O}(N\log(N))$[3] operations for sorting the decision values of the training and test sets. If the number of test and training observations differ we will use interpolation, while test data with a different prior distribution can be renormalized using the full LOO procedure.

Please note that the transformation relates to the distribution of the decision values, hence we do not consider nor estimate approximate posterior probabilities to implement the renormalization procedure, c.f., (Platt, 2000).

[3]This is the average time complexity of quicksort

Algorithm 1 Renormalization in SVM

Compute the decision values for training and test set: {we use LIBSVM (Chang and Lin, 2011)}
for $n = 1$ to N_{tr} **do**
  $d_n \leftarrow \sum_{i \in \mathcal{S}} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}_n) + b$
end for
for $n = 1$ to N_{te} **do**
  $g_n \leftarrow \sum_{i \in \mathcal{S}} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}_n) + b$
end for
$[\boldsymbol{d}_s, \ ] \leftarrow \text{sort}(\boldsymbol{d})$ {ascending order}
$[\ , I] \leftarrow \text{sort}(\boldsymbol{g})$ {ascending order}
for $n = 1$ to N_{te} **do**
  $y^h(\boldsymbol{x}_n) \leftarrow \boldsymbol{d}_s(I(n))$ {renormalized decision values of the test set, see Equation (10)}
end for

4. Experiments

To illustrate variance inflation and the effect of renormalization we examined the 18 real data sets listed in Table 2. The data sets are a combination of multidimensional data from the University of California, Irvine Machine Learning Repository (UCI) and gene expression microarray data from the Kent Ridge Biomedical Data Set Repository (KR).

4.1. Performance metrics

In imbalanced classification problems the standard accuracy measure tend to degenerate with increasing class imbalance. For example, a naive classifier that predicts all samples as the majority class has high accuracy, even though, it fails to classify any samples from the minority class correctly. Therefore, several other performance metrics combining the sensitivity (true positive rate) and the specificity (true negative rate) have been suggested for highly imbalanced classification tasks.The G-mean measure as proposed in (Kubat and Matwin, 1997) is the geometric mean of the specificity and sensitivity of the classifier, and given by

$$G = \sqrt{\text{sensitivity} \cdot \text{specificity}}. \tag{11}$$

Similarly, the Area Under ROC curve (ROC AUC) can be used to evaluate the balanced classification performance (Bradley, 1997). Using the threshold

produced by the SVM training, we report the G-mean, G, and the classifier accuracy, Acc, with and without applying the proposed renormalization scheme for all the experiments.

4.2. Results

For the multi-class data sets, we constructed a binary problem by classifying a random class with a sufficient number of samples versus all other samples. If a data set contained missing values, these were filled by interpolation. All data were Z-score normalized prior to classification to transform the features to a zero mean, unit variance form.

We performed experiments for decreasing non-linearity of the kernel embedding by letting $\frac{1}{\sqrt{\gamma}}$ vary from the 5th to the 80th percentile of the mutual distances of all observations in the training set. For each γ-level we performed 5-fold cross-validation to determine the slack parameter, C. Confidence intervals on the performance estimates where determined from 25 random splits of the data in disjoint training and test sets. The training and test data were evenly sized and with the same class distribution. Table 2 summarizes the results when using the histogram specification methods described in Equation (10) and Algorithm 1. For both the accuracy and the G-mean the best score is marked with boldface. For all but two data set applying the non-parametric renormalization is found to improve the G-mean whereas the effect on the accuracy measure is more ambiguous.

Figure 3 shows our findings on a subset of the data sets for varying kernel hyperparameter. In the top panel of Figure 3 performance is measured using accuracy while the lower panel shows the G-mean measure. In most of our experiments we found that renormalization improves performance. The results were most significant when using the G-mean measure. In several cases including normalization leads to a more non-linear optimal kernel embedding, suggesting the signal manifold is more non-linear than anticipated from standard SVM. Furthermore, renormalization often leads to results less sensitive to the choice of kernel parameter by maintaining good performance even for very non-linear kernel functions. Please refer to Figure A.1 through A.4 in Appendix A for similar figures for all data sets.

The histogram specification methods is seen to outperform all other approaches in most cases. However, learning the renormalization function thru the LOO scheme generally leads to similar performance. The orthogonal projection factor is found to lead to incremental differences compared to the

standard test error. Applying the projection factor and the histogram specification methods have similar time complexities, while the average speed-up achieved compared to the full LOO scheme is in the order 100. We therefore suggest to employ the histogram specification method in Eq. (10) when applicable, whereas the full LOO scheme can be applied at an increased computational cost, when the test data are known to follow a different prior distribution.

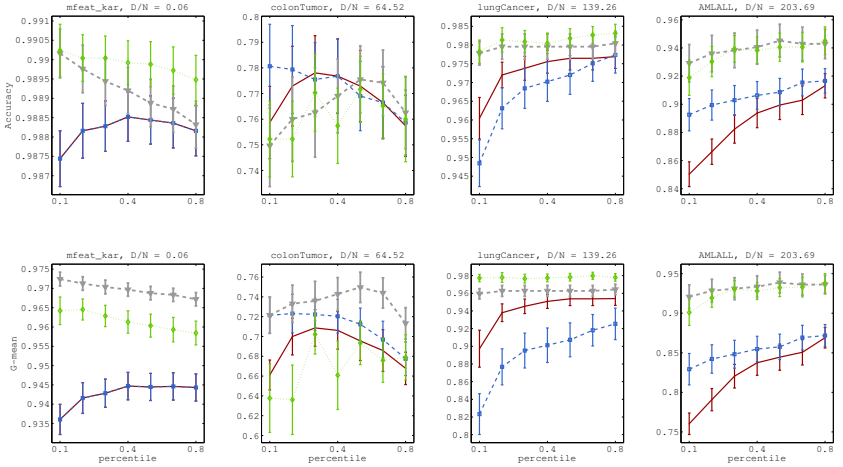


Figure 3: Mean performance measures ±1 std as a function of the non-linearity of the kernel embedding measured as percentiles of the mutual distances. Small percentiles correspond to very non-linear kernel embeddings. The top panel shows the accuracy while the bottom panel shows the G-mean. The figures summarize the standard test errors (red), the test error after including the lost orthogonal projection factor (see Eq. (9)) (blue), the renormalized test errors (see Eq. (10)) (gray), and the errors achieved by learning the renormalization function as described in Sec. 3.1 (green). It is evident that renormalization improves performance in most cases. Significant improvements are seen for the very non-linear regimes.

5. Conclusions

Supervised learning from small samples in high-dimensional spaces suffers from variance inflation potentially harming performance if variance inflation is not efficiently calibrated for. By an analytic result for simple regression

we saw that a contributing mechanism is the lost projection incurred as the training data only spans a subset of feature space. Variance inflation was shown to also affect the widely used radial basis function SVM. We have shown that performance may be improved by a simple non-parametric calibration scheme in many cases. Viability was tested on 18 benchmark data sets covering a wide range of N/D ratios.

Table 1: Overview of data sets used for the analysis. Data sources are the University of California, Irvine (UCI) Machine Learning Repository and the Kent Ridge Biomedical Data Set Repository (KR). N is the number of training samples, D is the number of features, and R is the imbalance ratio of the two classes. The four right most columns show the accuracy and G-mean with and without applying the histogram specification in Alg. 1 (denoted w and w/o respectively). Confidence intervals are given as ± 1 std. Boldface indicates the best performance. Renormalization is seen to improve the G-mean for most data sets, whereas the the improvement is more ambigious in terms of the accuracy measure.

Data set	N	D	R	C	Acc, w/o	Acc, w	G, w/o	G, w
Arcene, UCI	100	10000	0.44	1.00	$0.707(\pm 8e^{-3})$	$\mathbf{0.719(\pm 7e^{-3})}$	$0.701(\pm 1e^{-2})$	$\mathbf{0.716(\pm 7e^{-3})}$
Arrythmia, UCI[a b] (1)	225	279	0.46	5.47	$0.750(\pm 5e^{-3})$	$\mathbf{0.751(\pm 5e^{-3})}$	$0.734(\pm 5e^{-3})$	$\mathbf{0.739(\pm 5e^{-3})}$
Breast_w, UCI[b]	349	9	0.35	10.00	$0.969(\pm 2e^{-3})$	$\mathbf{0.970(\pm 2e^{-3})}$	$0.966(\pm 2e^{-3})$	$\mathbf{0.969(\pm 2e^{-3})}$
Habermann, UCI	152	3	0.27	3.21	$0.735(\pm 4e^{-3})$	$\mathbf{0.739(\pm 3e^{-3})}$	$0.428(\pm 2e^{-2})$	$\mathbf{0.453(\pm 9e^{-3})}$
Ionosphere, UCI	175	34	0.38	2.54	$0.944(\pm 3e^{-3})$	$\mathbf{0.946(\pm 3e^{-3})}$	$0.935(\pm 4e^{-3})$	$\mathbf{0.940(\pm 3e^{-3})}$
Mfeat_factors, UCI[a] (8)	1000	216	0.09	10.00	$0.995(\pm 5e^{-4})$	$\mathbf{0.995(\pm 5e^{-4})}$	$0.980(\pm 2e^{-3})$	$\mathbf{0.986(\pm 1e^{-3})}$
Mfeat_karhunen, UCI[a] (8)	1000	64	0.10	10.00	$0.989(\pm 6e^{-4})$	$\mathbf{0.990(\pm 6e^{-4})}$	$0.945(\pm 4e^{-3})$	$\mathbf{0.972(\pm 2e^{-3})}$
Mfeat_pixel, UCI[a] (8)	1000	240	0.11	3.21	$\mathbf{0.992(\pm 4e^{-4})}$	$0.991(\pm 6e^{-4})$	$0.959(\pm 2e^{-3})$	$\mathbf{0.976(\pm 2e^{-3})}$
Musk1, UC	237	166	0.44	35.21	$\mathbf{0.902(\pm 4e^{-3})}$	$0.899(\pm 3e^{-3})$	$\mathbf{0.901(\pm 4e^{-3})}$	$0.898(\pm 3e^{-3})$
Ozone_onehr, UCI[b]	1267	72	0.03	15.20	$\mathbf{0.971(\pm 3e^{-4})}$	$0.970(\pm 4e^{-4})$	$0.382(\pm 2e^{-2})$	$\mathbf{0.456(\pm 1e^{-2})}$
Parkinsons, UCI	97	22	0.23	7.07	$\mathbf{0.906(\pm 6e^{-3})}$	$0.898(\pm 6e^{-3})$	$0.843(\pm 1e^{-2})$	$\mathbf{0.860(\pm 9e^{-3})}$
Spambase, UCI	2300	57	0.39	6.09	$\mathbf{0.932(\pm 8e^{-4})}$	$0.931(\pm 8e^{-4})$	$0.925(\pm 9e^{-4})$	$\mathbf{0.925(\pm 8e^{-4})}$
Wdbc, UCI	284	30	0.36	4.50	$0.975(\pm 1e^{-3})$	$\mathbf{0.976(\pm 1e^{-3})}$	$0.970(\pm 2e^{-3})$	$\mathbf{0.971(\pm 1e^{-3})}$
AMLALL, KR	35	7129	0.29	3.16	$0.913(\pm 9e^{-3})$	$\mathbf{0.945(\pm 1e^{-2})}$	$0.869(\pm 1e^{-2})$	$\mathbf{0.939(\pm 1e^{-2})}$
ColonTumor, KR	31	2000	0.32	3.16	$\mathbf{0.778(\pm 1e^{-2})}$	$0.775(\pm 1e^{-2})$	$0.709(\pm 2e^{-2})$	$\mathbf{0.750(\pm 2e^{-2})}$
DLBCL_stanford, KR[b]	23	4026	0.48	1.31	$0.873(\pm 2e^{-2})$	$\mathbf{0.875(\pm 2e^{-2})}$	$0.869(\pm 2e^{-2})$	$\mathbf{0.875(\pm 2e^{-2})}$
LungCancer, KR	90	12533	0.50	7.07	$0.977(\pm 3e^{-3})$	$\mathbf{0.980(\pm 3e^{-3})}$	$0.954(\pm 8e^{-3})$	$\mathbf{0.964(\pm 6e^{-3})}$
Ovarian_61902, KR	126	15154	0.37	3.16	$\mathbf{0.989(\pm 2e^{-3})}$	$0.989(\pm 2e^{-3})$	$0.987(\pm 2e^{-3})$	$\mathbf{0.988(\pm 2e^{-3})}$

[a]Data set was originally multi-class. The class used as the minority class in the binary problem is given in parenthesis.
[b]Data set contains missing values.

Appendix A. Supplementary figures

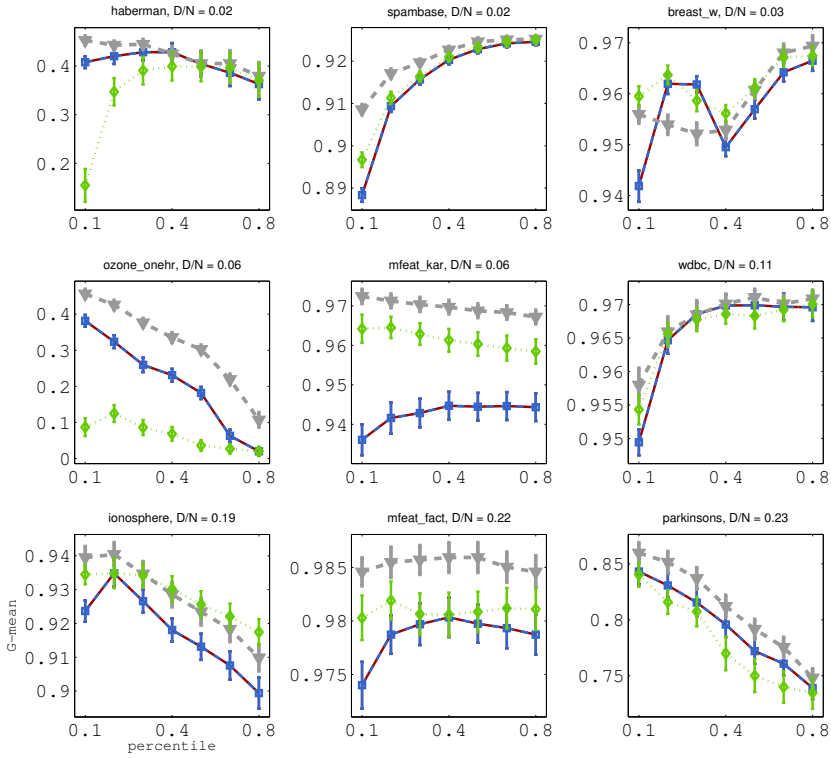


Figure A.1: The G-mean ±1 std as a function of the non-linearity of the kernel embedding measured as percentiles of the mutual distances. The figures summarize the standard test errors (red), the test error after including the lost orthogonal projection factor (see Eq. (9)) (blue), the renormalized test errors (see Eq. (10)) (gray), and the errors achieved by learning the renormalization function (green).

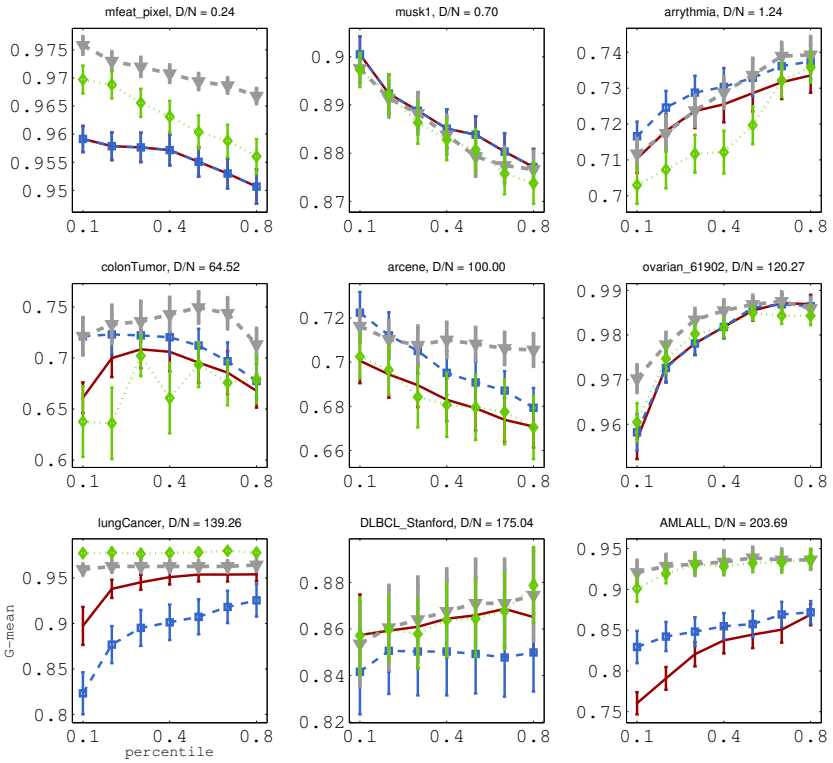


Figure A.2: The G-mean ±1 std as a function of the non-linearity of the kernel embedding measured as percentiles of the mutual distances. The figures summarize the standard test errors (red), the test error after including the lost orthogonal projection factor (see Eq. (9)) (blue), the renormalized test errors (see Eq. (10)) (gray), and the errors achieved by learning the renormalization function (green).

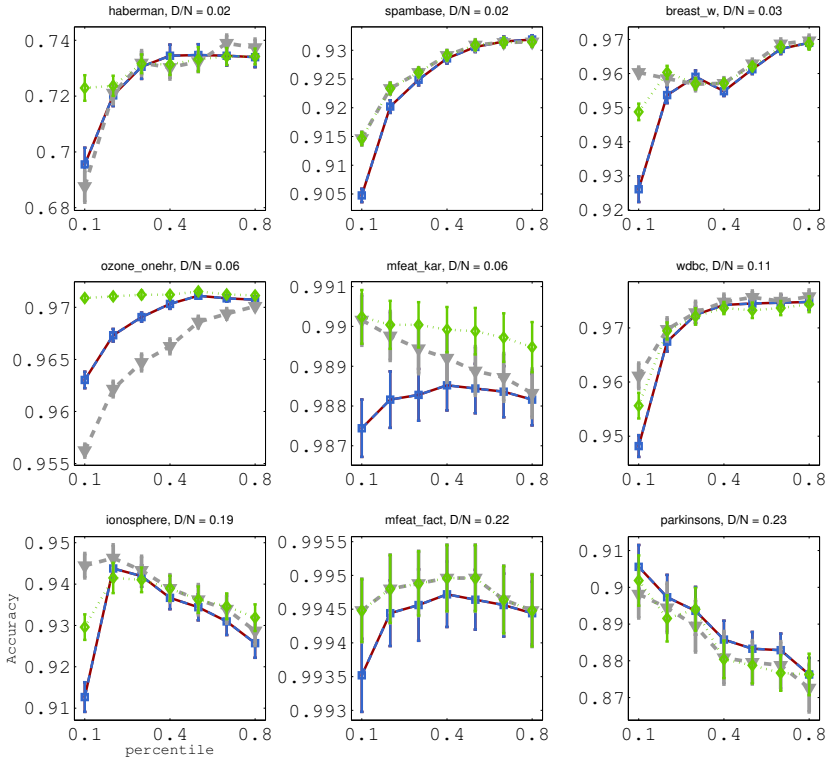


Figure A.3: The accuracy ±1 std as a function of the non-linearity of the kernel embedding measured as percentiles of the mutual distances. The figures summarize the standard test errors (red), the test error after including the lost orthogonal projection factor (see Eq. (9)) (blue), the renormalized test errors (see Eq. (10)) (gray), and the errors achieved by learning the renormalization function (green).

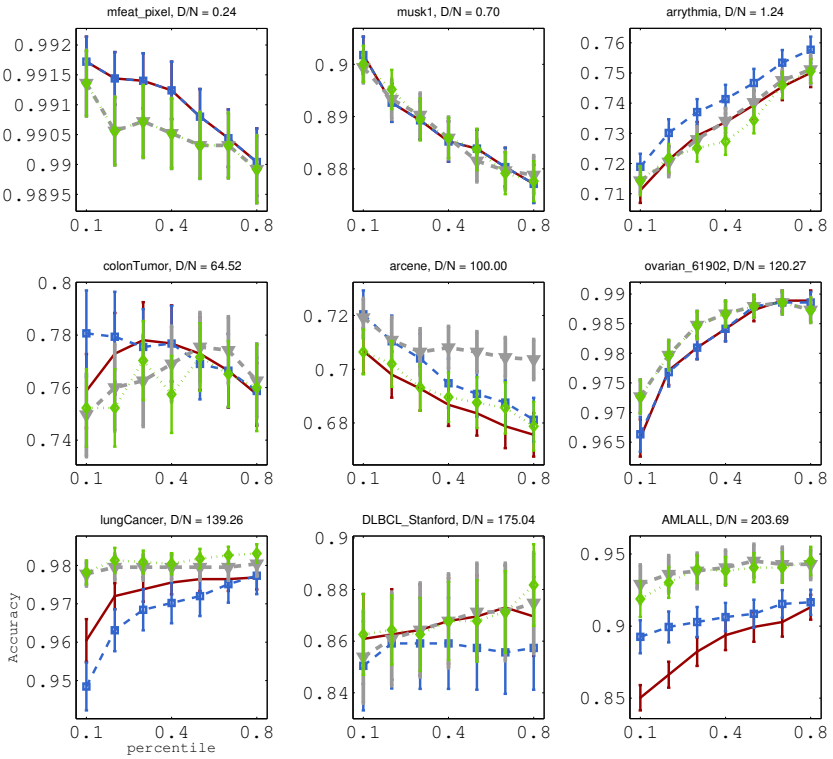


Figure A.4: The accuracy ±1 std as a function of the non-linearity of the kernel embedding measured as percentiles of the mutual distances. The figures summarize the standard test errors (red), the test error after including the lost orthogonal projection factor (see Eq. (9)) (blue), the renormalized test errors (see Eq. (10)) (gray), and the errors achieved by learning the renormalization function (green).

References

Abrahamsen, T. J., Hansen, L. K., 2011. A cure for variance inflation in high dimensional kernel principal component analysis. Journal of Machine Learning Research. 12, 2027–2044.

Abrahamsen, T. J., Hansen, L. K., 2012. Restoring the generalizability of svm based decoding in high dimensional neuroimage data. In: Langs, G., Rish, I., Grosse-Wentrup, M., Murphy, B. (Eds.), Machine Learning and Interpretation in Neuroimaging. Lecture Notes in Computer Science. Springer, pp. 256–263.

Barber, D., Saad, D., Sollich, P., 1995. Finite-size effects and optimal test set size in linear perceptrons. Journal of Physics (A) 28, 1325–1334.

Bradley, A. P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159.

Castro, C. L., Carvalho, M. A., Braga, A. P., Palmer-Brown, D., Draganova, C., Pimenidis, E., Mouratidis, H., 2009. An improved algorithm for SVMs classification of imbalanced data sets. Vol. 43. Springer Berlin Heidelberg, pp. 108–118.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

Christopher, C. M., 1996. Neural networks for pattern recognition. Oxford University Press, Oxford.

Frank, A., Asuncion, A., 2010. UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`

Gonzalez, R. C., Wintz, P., 1977. Digital image processing.

Hansen, L. K., 1993. Stochastic linear learning: Exact test and training error averages. Neural Networks 6 (3), 393–396.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning, Second Edition: Data Mining, Inference, and Prediction, 2nd Edition. Springer Series in Statistics. Springer.

Kimeldorf, G., Wahba, G., 1971. Some results on tchebycheffian spline functions. Journal of Mathematical Analysis and Applications 33 (1), 82–95.

Kjems, U., Hansen, L. K., Strother, S. C., 2001. Generalizable singular value decomposition for ill-posed datasets. In: Advances in Neural Information Processing Systems 13. MIT Press, pp. 549–555.

Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets: one-sided selection. In: Proc. 14th International Conference on Machine Learning. Morgan Kaufmann, pp. 179–186.

Penrose, R., Todd, J. A., 1956. On best approximate solutions of linear matrix equations. Mathematical Proceedings of the Cambridge Philosophical Society 52 (01), 17–19.

Platt, J. C., 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. Advances in Kernel Methods Support Vector Learning, 61–74.

Raskutti, B., Kowalczyk, A., Jun. 2004. Extreme re-balancing for svms: a case study. ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets 6 (1), 60–69.

Schölkopf, B., Smola, A. J., 2001. Learning with kernels: support vector machines, regularization, optimization and beyond. MIT Press.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., Krasser, S., 2009. SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (1), 281–288.

Vapnik, V. N., 1995. The nature of statistical learning. Springer-Verlag, New York.

Weiss, G. M., June 2004. Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets 6, 7–19.

Appendix H

A Randomized Heuristic for Kernel Parameter Selection with Large-Scale Multi-Class Data

Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. A Randomized Heuristic for Kernel Parameter Selection with Large-scale Multi-class Data. In *Proceedings of 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2011.
doi:10.1109/MLSP.2011.6064582.

2011 IEEE International Workshop on Machine Learning for Signal Processing
September 18-21, 2011, Beijing, China

A RANDOMIZED HEURISTIC FOR KERNEL PARAMETER SELECTION WITH LARGE-SCALE MULTI-CLASS DATA

Toke Jansen Hansen, Trine Julie Abrahamsen, Lars Kai Hansen

Section for Cognitive Systems
DTU Informatics
Technical University of Denmark

ABSTRACT

Over the past few years kernel methods have gained a tremendous amount of attention as existing linear algorithms can easily be extended to account for highly non-linear data in a computationally efficient manner. Unfortunately most kernels require careful tuning of intrinsic parameters to correctly model the distribution of the underlying data. For large-scale problems the multiplicative scaling in time complexity imposed by introducing free parameters in a cross-validation setup will prove computationally infeasible, often leaving pure ad-hoc estimates as the only option. In this contribution we investigate a novel randomized approach for kernel parameter selection in large-scale multi-class data. We fit a minimum enclosing ball to the class means in Reproducing Kernel Hilbert Spaces (RKHS), and use the radius as a quality measure of the space, defined by the kernel parameter. We apply the developed algorithm to a computer vision paradigm where the objective is to recognize 72.000 objects among 1.000 classes. Compared to other distance metrics in the RKHS we find that our randomized approach provides better results together with a highly competitive time complexity.

1. INTRODUCTION

Kernel based classification algorithms account for non-linearities in a computational sophisticated manner through use of the *kernel trick*. Robust selection of intrinsic kernel parameters involves a grid search combined with cross-validation (CV), but for large-scale multi-class data CV becomes both time consuming and resource intensive due to the multiplicative scaling in time complexity imposed by free parameters.

Only few attempts to specifically address the challenge of hyperparameter selection for multi-class problems have been made. While generic algorithms for choosing the hyperparameter in multi-class Support Vector Machines (SVM) was suggested in [1], both [2] and [3] aimed at merely reducing the number of train-validation cycles, e.g., by performing CV on a subsample of the data prior to a restricted line search on the full data set. Several other attempts to more computational attractive approximations to K-fold CV have been made for binary classification [4, 5, 6]. However, in [7] it was shown that all of these approximation schemes were inferior to 5-fold CV.

In this contribution we exploit that previous studies on binary classification have shown, how the intercluster distance in feature space and the optimal hyperparameter defining the RKHS correlates [8, 9]. We extend these attempts to multi-class problems where heuristics for good class separation becomes less immediate. In previous work on intercluster distance based measures for choosing the hyperparameter, it was briefly suggested to maximize the mean of the intercluster distances for multi-class problems [8, 9].

We propose a novel algorithm for hyperparameter selection where a Minimum Enclosing Ball (MEB) is used as a measure of the dispersion of cluster means in the RKHS. Hence, we seek the RKHS that maximizes the MEB. A sublinear algorithm for finding the MEB in a finite dimensional input space was introduced by [10]. In this paper, we device a randomized approximation for MEB estimation in the infinite dimensional RKHS, thereby providing competitive time complexities with respect to existing distance metrics in the RKHS. We demonstrate the developed algorithm by considering image classification on the Amsterdam Library of Object Images (ALOI) [11] and compare the performance with the *median*, *mean*, *maximum* and *minimum* distance measures in the RKHS.

In our experiments we focus on the Gaussian kernel, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$, and use a multi-class SVM in a 5-fold CV setting to establish a ground-truth estimate for comparison with the heuristics. However, the developed heuristics trivially generalize to other kernel functions and kernel machines.

This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

978-1-4577-1623-2/11/$26.00

1.1. Organization

This paper is organized in the following way: In Section 2 we introduce the theory behind kernel machines together with the derivation of the considered MEB algorithms. In Section 3 we apply the MEB heuristics to a multi-class object recognition problem and compare our novel approach with related heuristics. Finally, Section 4 briefly concludes the paper.

2. THEORY

Let $\mathcal{H}$ be the RKHS associated with the kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \varphi(\boldsymbol{x})^T \varphi(\boldsymbol{x}')$ (see notation[1]), where $\varphi : \mathcal{X} \mapsto \mathcal{H}$ is a possibly non-linear map from the $D_{\mathcal{X}}$-dimensional input space, $\mathcal{X}$, to the $D_{\mathcal{H}}$-dimensional feature space, $\mathcal{H}$, (possibly infinite dimensional). This is known as the *kernel trick* which states that innerproducts in $\mathcal{H}$ can be computed in terms of kernel evaluations in $\mathcal{X}$. For convenience, all kernel evaluations are collected in the kernel matrix, $\boldsymbol{K} \in \mathbb{R}^{N \times N}$.

2.1. Kernel Machines for Classification

Given a set of training data $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, $\boldsymbol{x} \in \mathbb{R}^{D_{\mathcal{X}}}$, $y \in \{1, -1\}$, the SVM loss function can be expressed as follows[2]:

$$L_{\text{SVM}} = \min_{\boldsymbol{w} \in \mathcal{H}} \sum_{i=1}^{N} \max(0, 1 - y_i \boldsymbol{w}^\top \varphi(\boldsymbol{x}_i)) + \lambda \|\boldsymbol{w}\|^2, \quad (1)$$

where the first term corresponds to the Hinge-loss function. A computational benefit of the SVM is that classification is based on a subset of training samples defining the margin, these samples are also known as *support vectors*. Note that the slack term controlling the width of the margin is given by $C = \frac{1}{\lambda}$. By applying the representer theorem to rewrite $\boldsymbol{w}$ as $\boldsymbol{w} = \sum_{i=1}^N \alpha_i \varphi(\boldsymbol{x}_i)$, the kernel trick is made applicable. For all kernel type machines the decision function can be expressed as a linear combination of kernel evaluations.

Often direct optimization of the hyperparameters will be infeasible due to non-convexities introduced by the kernel, and a Bayesian treatment such as Automatic Relevance Determination (ARD) will prove computationally heavy even for moderate sized problems when analytic integration over the parameter space is intractable.

For binary classifiers such as the loss functions stated in Eq. (1), two general schemes can be applied to accommodate for multiple classes. One approach is to build one-versus-rest classifiers and to choose the class which classifies the test point with greatest margin/probability. Another strategy is to build a set of one-versus-one classifiers, and select the class based on majority voting [12]. This scheme is applied in our experiments. Even though more classifiers must be trained, the latter approach may prove faster, since the training data set for each classifier is much smaller. The immediate advantage of multiple binary classifiers is that averaging over the classifier decisions will most likely reduce the variance.

[1] Bold uppercase letters denote matrices, bold lowercase letters represent column vectors, and non-bold letters denote scalars.

[2] The bias term, b, has been omitted for notational convenience.

2.2. Clustering Geometry in RKHSs

Given a K class problem with N observations, the j^{th} cluster mean in the RKHS is given by:

$$\boldsymbol{m}_j = \frac{1}{N_{\mathcal{S}_j}} \sum_{i \in \mathcal{S}_j} \varphi(\boldsymbol{x}_i),$$

where $\mathcal{S}_j$ denotes the set of observations belonging to class j. For small values of γ relative to the length scale in input space, any kernelized method approach the equivalent linear method because high order terms in the taylor expansion of the RBF kernel becomes insignificant. Hence, in order to account for non-linearities in the data, γ should be increased. However, in the limit, $\gamma \to \infty$, the following holds:

$$\lim_{\gamma \to \infty} \boldsymbol{K} = \boldsymbol{I} \Rightarrow \lim_{\gamma \to \infty} ||\boldsymbol{m}_i - \boldsymbol{m}_j||^2 = \frac{1}{N_{\mathcal{S}_i}} + \frac{1}{N_{\mathcal{S}_j}}.$$

This result implies that all observations become uncorrelated and the mean of each class will approach $\boldsymbol{0}$ at a rate inversely proportional to the number of samples within that class. A further result of $\boldsymbol{K}$ approaching $\boldsymbol{I}$ is that the variance of any partitioning of the observations approaches 1, making signal extraction infeasible. Decreasing γ from ∞ will introduce off-diagonal contributions in $\boldsymbol{K}$, leading to an increased distance between the means when the cluster assumption holds (cf. [13]). Since the emerging off-diagonal elements of $\boldsymbol{K}$ depend on the distribution in input space, the exact structure of the intercluster distance as a function of γ will be difficult to quantify without an explicit search, hence finding the optimal γ is nontrivial.

2.3. Minimum Enclosing Ball

Let $\boldsymbol{A}$ denote the matrix of cluster means in feature space:

$$\boldsymbol{A} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_K],$$

where $\boldsymbol{A} \in \mathbb{R}^{D_{\mathcal{H}} \times K}$. As a measure of the class separation in $\mathcal{H}$ we fit a minimum enclosing ball to the cluster means and choose the optimal γ, and thereby RKHS, as the one leading to the largest MEB (see Figure 1). The MEB problem can be formulated as finding the smallest Euclidean ball

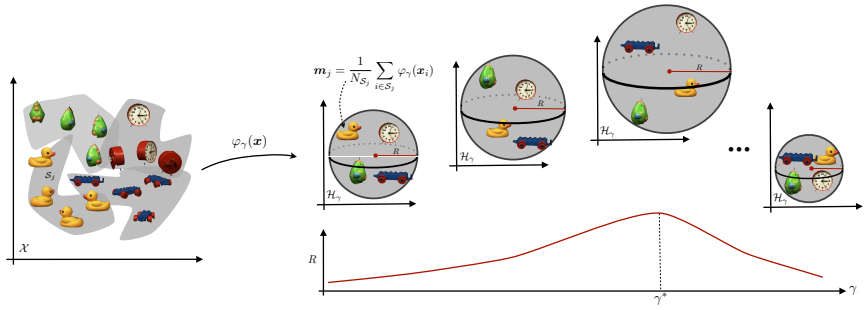


Fig. 1. Illustration of the minimum enclosing ball approach to hyperparameter selection. Each instance is mapped from input space, $\mathcal{X}$, to a RKHS defined by the hyperparameter, γ. The class means are calculated in the RKHS, and the radius, R, of the smallest ball that encloses all cluster means are determined using either the MEB, RMEB or R2MEB algorithm. The optimal hyperparameter is chosen as the one which maximizes the radius of the minimum enclosing ball.

in $D_{\mathcal{H}}$ which contains all columns of $\boldsymbol{A}$, which can be formulated as:

$$\boldsymbol{c}^* = \underset{\varphi(\boldsymbol{x}) \in \mathbb{R}^{D_{\mathcal{H}}}}{\operatorname{argmin}} \max_{i \in [K]} ||\varphi(\boldsymbol{x}) - \boldsymbol{a}_i||^2,$$

where $\max_{i \in [K]} ||\varphi(\boldsymbol{x}) - \boldsymbol{a}_i||^2$ is the radius of the ball, and $\boldsymbol{c}^*$ is the center which minimizes the ball. The above can be reformulated as [10]:

$$\boldsymbol{c}^* = \underset{\varphi(\boldsymbol{x}) \in \mathbb{R}^{D_{\mathcal{H}}}}{\operatorname{argmin}} \max_{\boldsymbol{p} \in \Delta_K} \sum_{i \in [K]} p_i ||\varphi(\boldsymbol{x}) - \boldsymbol{a}_i||^2 \,, \quad (2)$$

where $\Delta_K = \{\boldsymbol{p} \in \mathbb{R}^K | \sum_i p_i = 1,\ p_i \geq 0\}$ is the unit simplex. Thus, maximizing $\boldsymbol{p}$ puts all its weight on the farthest point.

Since we are only interested in finding the radius of the ball, the possible infinite dimensionality of $\boldsymbol{c}^*$ is not of importance. To calculate the distance $\|\varphi(\boldsymbol{x}) - \boldsymbol{a}_i\|^2$ we follow [10] and substitute $\varphi(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{p}$ leading to:

$$\|\boldsymbol{A}\boldsymbol{p} - \boldsymbol{a}_i\|^2 = \boldsymbol{p}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{p} + \boldsymbol{a}_i^\top \boldsymbol{a}_i - 2\boldsymbol{p}^\top \boldsymbol{A}^\top \boldsymbol{a}_i \,, \quad (3)$$

where the two terms ($\boldsymbol{a}_i^\top \boldsymbol{a}_i$ and $\boldsymbol{A}^\top \boldsymbol{a}_i$) are simply subsets of the full matrix $\boldsymbol{A}^\top \boldsymbol{A}$. Next we apply the kernel trick:

$$\begin{aligned}(\boldsymbol{A}^\top \boldsymbol{A})_{m,n} &= \frac{1}{N_{\mathcal{S}_m} N_{\mathcal{S}_n}} \sum_{i \in \mathcal{S}_m} \sum_{j \in \mathcal{S}_n} \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j) \\ &= \frac{1}{N_{\mathcal{S}_m} N_{\mathcal{S}_n}} \sum_{i \in \mathcal{S}_m} \sum_{j \in \mathcal{S}_n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \,. \quad (4)\end{aligned}$$

The dual problem of Eq. (2) can now be used to derive a primal-dual algorithm for optimizing Eq. (2) (cf. [10]).

The algorithm is shown in Alg. 1, where T is the desired number of optimization steps. By running the algorithm for various values of γ, the optimal hyperparameter can be found as $\gamma^* = \underset{\gamma \in \mathbb{R}}{\operatorname{argmax}}\ R(\gamma)$, where R is the radius of the MEB in $\mathcal{H}$.

Algorithm 1 Primal-dual MEB algorithm

1: Let $\boldsymbol{q}^0 \leftarrow [\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}]^\top$ and $\boldsymbol{p}^0 \leftarrow \boldsymbol{q}^0$
2: **for** $t = 1$ to T **do**
3: **for all** $i \in [K]$ **do**
4: $q_i^t = q_i^{t-1} \exp(-||\boldsymbol{A}\boldsymbol{p}^{t-1} - \boldsymbol{a}_i||^2)$ {Eq. (3)-(4)}
5: **end for**
6: $\boldsymbol{q}^t = \boldsymbol{q}^t / ||\boldsymbol{q}^t||$
7: $\boldsymbol{p}^t = (1 - 1/t)\boldsymbol{q}^{t-1} + 1/t \cdot \boldsymbol{q}^t$
8: **end for**
9: $k = \text{argmax}\ \boldsymbol{q}^T$
10: $R = ||\boldsymbol{A}\boldsymbol{p}^T - \boldsymbol{a}_k||^2$

When deriving the algorithm we exploit that the point corresponding to the largest value of $\boldsymbol{q}$ is the one farthest from the center. Hence, the radius of the ball can be found as the distance between the estimated center, $\boldsymbol{c}^* \approx \boldsymbol{A}\boldsymbol{p}^T$, and the farthest cluster mean, $\boldsymbol{a}_k$. In line 7, we assume that $\boldsymbol{A}$ is invertible, which translates to requiring that the K cluster means span a K-dimensional subspace of $\mathcal{H}$. If $\boldsymbol{A}$ is degenerate, line 7 becomes an approximation. However, since the columns of $\boldsymbol{A}$ are constructed as linear combinations of the φ-mapped observations this is a fair assumption, since for any positive definite kernel all $\varphi(\boldsymbol{x}_i)$'s are linearly independent as long as $\boldsymbol{x}_i = \boldsymbol{x}_j$ iff $i = j$.

In order to reduce the time complexity, the primal-dual

algorithm can be randomized [10]. Instead of calculating $\boldsymbol{Ap}$, we sample index $j \in [K]$ with probability p_j and substitute $\boldsymbol{Ap}$ with $\boldsymbol{a}_j$. Since j is chosen randomly, $\boldsymbol{a}_j$ is an unbiased estimator of $\boldsymbol{Ap}$. The randomized MEB (RMEB) algorithm is given in Alg. 2.

Algorithm 2 RMEB algortihm

1: Let $\boldsymbol{q}^0 \leftarrow [\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}]^\top$ and $\boldsymbol{p}^0 \leftarrow \boldsymbol{q}^0$
2: **for** $t = 1$ to T **do**
3: **for all** $i \in [K]$ **do**
4: Choose $j^{t-1} \in [K]$ by $j^{t-1} \leftarrow j$ with prob. p_i^{t-1}
5: $q_i^t = q_i^{t-1} \exp(-||\boldsymbol{a}_{j_{t-1}} - \boldsymbol{a}_i||^2)$
6: **end for**
7: $\boldsymbol{q}^t = \boldsymbol{q}^t / ||\boldsymbol{q}^t||$
8: $\boldsymbol{p}^t = (1 - 1/t)\boldsymbol{q}^{t-1} + 1/t\boldsymbol{q}^t$
9: **end for**
10: $k = \text{argmax}\ \boldsymbol{q}^T$
11: $R = ||\boldsymbol{Ap}^T - \boldsymbol{a}_k||^2$

The time complexity can be reduced further by randomizing the radius estimate in Line 11 of Alg. 2, leading to the randomized radius MEB (R2MEB) algorithm, where the full $\boldsymbol{Ap}^T$ is approximated by choosing index $l \in [K]$ with probability p_l^T and the radius is then estimated as $R = ||\boldsymbol{A}p_l^T - \boldsymbol{a}_k||^2$. For large-scale problems randomization of the radius estimate will lead to a significant speed up, as only one column of $\boldsymbol{Ap}^T$ is calculated. Since this randomization approximates the center of the MEB by a single data point, some variability of the estimate is inevitable, however, by construction the estimate converges in expectation.

Evidently, the radius of the MEB is upperbounded by half of the maximum of the pairwise distances between the cluster means.

2.4. Other Heuristics for Measuring Class Dispersion

Other natural measures of the class seperation include the *minimum, mean, median* or *maximum* of the pairwise distances between the cluster means which can all be found based on the kernel matrix $\boldsymbol{A}^\top \boldsymbol{A}$.

However, these heuristics all suffer from instability in different scenarios. If two cluster means are located very close in the RKHS, the class dispersion measure based on maximizing the minimum distance will collapse. In this case the MEB approach is still robust as long as only a small fraction of the classes are "very close", i.e., the pseudo-inverse of $\boldsymbol{A}$ is still well-defined. In the other extreme, if one class is very distinct from the rest, maximizing the maximum pairwise distance, are not guaranteed to provide good class separation of the remaining classes. However, in this case the MEB approach will not only separate the "odd" class but also optimize the class separation of the more "similar" classes in order to achieve the largest possible minimum enclosing ball. While both the mean and the median is slightly more robust in such scenarios, it will still fail in extreme cases. Finally, a potential issue regarding the median measure is that it allows for large variability in the distances as long as the median distance do not change, thereby not necessarily identifying the optimal class separation for all classes.

In terms of time complexity the considered methods all require $\mathcal{O}(N^2)$ operations for computing the Gram matrix[3], but only $\mathcal{O}(K^2)$ storage since only intercluster distances between class means in the RKHS are required by the heuristics, i.e., we only store $\boldsymbol{A}^\top \boldsymbol{A}$. Since the *mean*, *maximum* and *minimum* distances can be updated in an online fashion, these can be calculated on the fly using simple bookkeeping while computing $\boldsymbol{A}^\top \boldsymbol{A}$, hence, these only add a constant term to the overall time complexity. On the other hand there exist no $\mathcal{O}(1)$ online update for the *median*, so this quantity requires an additional $\mathcal{O}(K \cdot \log(K))$[4] for sorting the distances. For the MEB algorithm the additional time complexity becomes $\mathcal{O}(T \cdot K^2)$ due to the matrix operations in the inner loop of Alg. 1, whereas the RMEB can be computed in $\mathcal{O}(T \cdot K)$. According to our results sufficient convergence was achieved using $T < \log(K)$ iterations, i.e., the RMEB has an average lower time complexity than the *median*. However, for all aforementioned approaches $\mathcal{O}(N^2)$ remains the dominating factor. The coarse approximation exploited by the R2MEB algorithm can further reduce the overall time complexity considerably. Assuming that the classes are fairly balanced, i.e., each class will contain approximately $\frac{N}{K}$ samples, the time complexity for computing the Gram matrix is reduced to $\mathcal{O}(\frac{\min(T,K)}{K} \cdot N^2)$ since elements can be cached and computed in a lazy fashion. Also for this implementation our results show that reasonable results are obtained for $T < \log(K)$.

3. EXPERIMENTS

For efficient computation of all of the heuristics we precompute the tensor $(\boldsymbol{A}^\top \boldsymbol{A})_\gamma \in \mathbb{R}^{K \times K \times \Gamma}$, where the third dimension corresponds to different choices of the hyperparameter, γ, in the Gaussian kernel, selected from the range $\gamma \in \{2^{-6}, 2^{-5}, \ldots, 2^5\}$. The performance of the heuristics are evaluated using image features distorted by uncorrelated gaussian noise having standard deviation $\sigma \in \{0.0, 0.1, 0.2\}$. We compare the heuristics against the multi-class SVM implementation found in LIBSVM [12], where 5-fold cross-validation is applied to obtain a ground truth estimate of both γ and the slack value, $C \in \{10^{-5}, 10^{-4}, \ldots, 10^{10}\}$.

[3] For simplicity we ignore symmetry of $\boldsymbol{A}^\top \boldsymbol{A}$

[4] This is the average time complexity of quicksort.

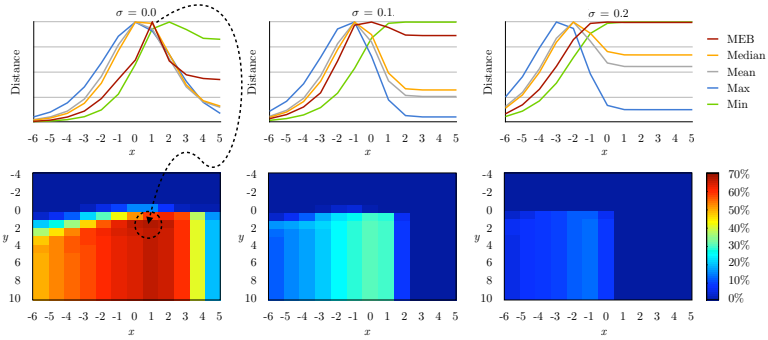


Fig. 2. The panels left to right show the different noise levels ($\sigma = 0.0$, $\sigma = 0.1$ and $\sigma = 0.2$ respectively). The upper panel shows the median, mean, min, and max distances between the class means as well as the radius of the MEB for varying γ values. The results have been normalized for easier comparison. The lower panel shows the 5-fold cross validation accuracy. The parameters are given as $C = 10^y$ and $\gamma = 2^x$. For $\sigma = 0.00$ and $\sigma = 0.1$ it is evident that only the MEB approach peaks at the optimal γ as seen from the CV plot. For $\sigma = 0.2$ the MEB approach saturates at optimal γ value, but no well-defined peak occurs.

3.1. The Amsterdam Library of Object Images

The Amsterdam Library of Object Images (ALOI) is a collection of 1.000 objects that have been recorded for scientific purposes [11]. In the present we consider object classification where the object viewpoint is shifted in steps of 5° yielding a total of 72 images of each object. For each image in the dataset we compute a set of Speeded Up Robust Features (SURF) inspired by the Scale Invariant Feature Transform (SIFT). Both are used to detect and describe local features in images [14, 15]. Since the number of extracted features may vary across the considered images we apply principal component analysis (PCA) to the extracted features of an image, and select the first principal axis to represent the entire image in a compact low dimensional representation.

Even though all of the heuristics can be computed easily for the entire dataset, we restrict our analysis to a subset of 100 objects from the library. This is necessary, since the establishment of a ground-truth estimate by complete SVM CV proves computationally infeasible for more classes.

The results are summarized in Figure 2 and Table 1. Figure 2 shows the results from the heuristics for varying γ as well as the CV results for comparison.The lower panel clearly shows that the performance is very sensitive to the choice of γ. For the three MEB approaches we use $T = 5$ to obtain an algorithm (RMEB) with a strictly lower time complexity than the median. For illustrative purposes we only show the non-randomized MEB approach in Fig. 2. Table 1 show the classification accuracy of the SVM when using the hyperparameter selected by the various heuristics. It is evident that using the MEB approaches for hyperparameter selection leads to better classification for all noise levels.

When comparing the location of the peaks of the class-separation-measures in the upper panel of Figure 2 to the CV results from the SVM in the lower panel, it is evident that using the minimum intercluster distance for hyperparameter selection leads to too large γ-values, while the other standard heuristics all suggest a too small hyperparameter and thereby too linear kernel embeddings. On the contrary, the MEB approach identifies the optimal γ-value for both $\sigma = 0.0$ and $\sigma = 0.1$. In the very noisy setting ($\sigma = 0.2$) the MEB estimate saturates around the optimal γ-value but no significant peak occurs. However, by Occam's razor one could argue to choose the simplest model (smallest γ in this case). Interestingly, the RMEB algorithm is found to actually peak at γ^* for $\sigma = 0.2$, thereby yielding a better classification accuracy than the MEB implementation in this case. This could be caused by different convergence characteristics of the two algorithms.

4. CONCLUSIONS

We have shown how maximizing the radius of the minimum enclosing ball of the cluster means in the RKHS provide a

Noise level	MEB	RMEB	R2MEB	Median	Mean	Max	Min
$\sigma = 0.0$	**67.11 %**	66.04 % (1.18)	65.03 % (2.50)	65.83 %	65.83 %	65.83 %	65.83 %
$\sigma = 0.1$	**31.25 %**	29.58 % (1.88)	28.32 % (2.78)	28.65 %	28.65 %	28.65 %	9.71 %
$\sigma = 0.2$	8.82 %	**12.49 %** (1.33)	11.61 % (1.39)	12.38 %	12.38 %	10.85 %	0.71 %

Table 1. The table shows the classification accuracy of the SVM when using the hyperparameter suggested by the various heuristics on the ALOI dataset for various noise levels. For the three MEB methods five optimization steps were taken ($T = 5$) and for the randomized approaches the standard deviation is given in brackets. For each noise level, the best classification rate is marked in bold. Clearly the MEB approaches lead to more optimal γ-values and thereby higher accuracy of the SVM.

meaningful heuristic for finding the optimal hyperparameter (and hence, RKHS) for kernel machines in multi-class classification problems. Compared to other standard distance metrics in RKHSs we found that the MEB approach provides better results together with a highly competitive time complexity for large scale multi-class data. Under noisy conditions, the performance of the randomized MEB approach indicated a faster convergence of the RMEB than the MEB approach in this setting.

Due to the low time complexity and improved performance, we suggest to use the minimum enclosing ball for crude hyperparameter selection in large-scale problems.

Future work includes testing on a wider range of large scale multi-class classification problems. Furthermore, outlier detection by fitting a MEB in the RKHS is a natural unsupervised extension.

5. REFERENCES

[1] Ana Carolina Lorena and André C. P. L. F. de Carvalho, "Evolutionary tuning of svm parameter values in multiclass problems," *Neurocomput.*, vol. 71, pp. 3326–3334, October 2008.

[2] Matthias Varewyck and Jean-Pierre Martens, "A Practical Approach to Model Selection for Support Vector Machines With a Gaussian Kernel," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 330–340, Apr. 2011.

[3] CJ Van Heerden and E. Barnard, "Towards understanding the influence of svm hyperparameters," in *21st Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2010, pp. 69–74.

[4] Thorsten Joachims, *The Maximum-Margin Approach to Learning Text Classifiers:Methods, theory, and algorithms*, Ph.D. thesis, Dortmund University, 2001.

[5] Grace Wahba, *Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV*, pp. 69–88, MIT Press, Cambridge, MA, USA, 1999.

[6] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Comput.*, vol. 12, pp. 2013–2036, September 2000.

[7] Kaibo Duan, S. Sathiya Keerthi, and Aun Neow Poo, "Evaluation of simple performance measures for tuning svm hyperparameters," *Neurocomputing*, vol. 51, pp. 41–59, 2003.

[8] Kuo-Ping Wu and Sheng-De Wang, "Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space," *Pattern Recognition*, vol. 42, pp. 710–717, May 2009.

[9] Song Xiaoshan, Jiang Xiaoyu, Han Chongzhao, and Luo Jianhua, "Inter-class distance based kernel parameter evaluating method for rbf-svm," *Digital Manufacturing and Automation, International Conference on*, vol. 1, pp. 853–858, 2010.

[10] Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff, "Sublinear optimization for machine learning," in *FOCS*, 2010, pp. 449–457.

[11] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[12] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf, "Cluster kernels for semi-supervised learning," in *Neural Information Processing Systems*, 2002, pp. 585–592.

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.

[15] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004.

Bibliography

[ABR64] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[Abr09] Trine Julie Abrahamsen. Kernel methods for de-noising with neuroimaging application. Master's thesis, Technical University of Denmark, Department of Informatics and Mathematical Modeling, Intelligent Signal Processing, Kgs. Lyngby, Denmark, 2009.

[ACS+11] L.E. Aik, T.W. Choon, I. Soesanti, A. Susanto, T.S. Widodo, M. Tjokronagoro, S. Zaman, W. Slany, S.N. Ahsan, F. Hyder, et al. Enhancing passive stereo face recognition using pca and fuzzy c-means clustering. *International Journal of Video & Image Processing and Network Security*, 12(4):12–22, 2011.

[AGPCVH13] Jerónimo Arenas-Garcıa, Kaare Brandt Petersen, Gustavo Camps-Valls, and Lars Kai Hansen. Kernel multivariate analysis framework for supervised subspace learning. *IEEE Signal Processing Magazine*, 30(4):16–29, 2013.

[AH09] Trine Julie Abrahamsen and Lars Kai Hansen. Input space regularization stabilizes pre-images for kernel pca de-noising. In *Proceedings of 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2009.

[AH10] Trine Julie Abrahamsen and Lars Kai Hansen. Sparse nonlinear denoising of fmri: Performance and pattern reproducibility.

In *Neural Information Processing Systems (NIPS) Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions*, 2010.

[AH11a] Trine Julie Abrahamsen and Lars Kai Hansen. A cure for variance inflation in high dimensional kernel principal component analysis. *Journal of Machine Learning Research*, 12:2027–2044, 2011.

[AH11b] Trine Julie Abrahamsen and Lars Kai Hansen. Regularized pre-image estimation for kernel pca de-noising. *Journal of Signal Processing Systems*, 65:403–412, 2011.

[AH11c] Trine Julie Abrahamsen and Lars Kai Hansen. Sparse non-linear denoising: Generalization performance and pattern reproducibility in functional mri. *Pattern Recognition Letters*, 32(15):2080 – 2085, 2011.

[AH12] Trine Julie Abrahamsen and Lars Kai Hansen. Restoring the generalizability of svm based decoding in high dimensional neuroimage data. In Georg Langs, Irina Rish, Moritz Grosse-Wentrup, and Brian Murphy, editors, *Machine Learning and Interpretation in Neuroimaging*, Lecture Notes in Computer Science, pages 256–263. Springer Berlin Heidelberg, 2012.

[AH13] Trine Julie Abrahamsen and Lars Kai Hansen. Variance inflation in high dimensional support vector machines. *Pattern Recognition Letters*, Accepted, 2013.

[AKJ04] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. *Machine Learning: ECML 2004*, pages 39–50, 2004.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc*, 68(3):337–404, 1950.

[ARS07] Pablo Arias, Gregory Randall, and Guillermo Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 18-23 jun 2007.

[AY01] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. *ACM Sigmod Record*, 30(2):37–46, 2001.

[BBB66] Richard Bellman, Richard Ernest Bellman, and Richard Ernest Bellman. *Adaptive control processes: A guided tour*. Princeton University Press, 1966.

[BBM08] Mikio L Braun, Joachim M Buhmann, and Klaus-Robert Müller. On relevant dimensions in kernel feature spaces. *The Journal of Machine Learning Research*, 9:1875–1908, 2008.

[BBZ07] Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007.

[BEWB05] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research*, 6:1579–1619, 2005.

[BGRS99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? *Database Theory - ICDT 1999*, pages 217–235, 1999.

[BGV92] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[BHH+13] Daniel Bartz, Kerr Hatrick, Christian W. Hesse, Klaus-Robert Müller, and Steven Lemm. Directional variance adjustment: Bias reduction in covariance matrices based on factor analysis with an application to portfolio optimization. *PLoS ONE*, 8(7), 2013.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 2006.

[BN03] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[BPV+04] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16:177–184, 2004.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[BWS04] Gökhan H. Bakir, Jason Weston, and Bernhard Schölkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems 16*, pages 449–456. MIT Press, 2004.

[BZT04] G. H. Bakir, Alexander Zien, and Koji Tsuda. Learning to find graph pre-images. In *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 253–261. Springer Berlin / Heidelberg, 2004.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.

[CC00] T.F. Cox and M.A.A. Cox. *Multidimensional scaling*, volume 88. Chapman & Hall/CRC, 2000.

[CCB09] Cristiano Leite Castro, Mateus Araujo Carvalho, and Antônio Padua Braga. An improved algorithm for svms classification of imbalanced data sets. *Engineering Applications of Neural Networks*, pages 108–118, 2009.

[CHW10] Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. In *FOCS*, pages 449–457, 2010.

[CLI+11] C. Chen, F. Lauze, C. Igel, A. Feragen, M. Loog, and M. Nielsen. Towards exaggerated image stereotypes. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 422–426. IEEE, 2011.

[CSL+12] Chen Chen, L Sørensen, F Lauze, Christian Igel, Marco Loog, Aasa Feragen, Marleen de Bruijne, and Mads Nielsen. Towards exaggerated emphysema stereotypes. In *SPIE Medical Imaging*, pages 83150Q–83150Q. International Society for Optics and Photonics, 2012.

[CST00] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1):131–159, 2002.

[DBK+97] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, pages 155–161, 1997.

[DK05] Kai-Bo Duan and Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. *Multiple Classifier Systems*, pages 732–760, 2005.

[DKP03] Kaibo Duan, S. Sathiya Keerthi, and Aun Neow Poo. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41–59, 2003.

[FA10] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[Fle81] R. Fletcher. Practical methods of optimization: Vol. 2: Constrained optimization. *JOHN WILEY & SONS, INC., ONE WILEY DR., SOMERSET, N. J. 08873, 1981, 224*, 1981.

[Fri08] J. Friedman. Fast sparse regression and classification. *T echnicalreport, Standford Uni*, 2008.

[FWV07] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *Knowledge and Data Engineering, IEEE Transactions on*, 19(7):873–886, 2007.

[FWXT13] Zizhu Fan, Jinghua Wang, Baogen Xu, and Pengzhi Tang. An efficient kpca algorithm based on feature correlation evaluation. *Neural Computing and Applications*, pages 1–12, 2013.

[GA11] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[GBS05] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

[GGvM89] Walter Gander, Gene H Golub, and Urs von Matt. A constrained eigenvalue problem. *Linear Algebra and its applications*, 114:815–839, 1989.

[GMARH12] P. Garcia-Moreno, A. Artes-Rodriguez, and L.K. Hansen. A hold-out method to correct pca variance inflation. In *Cognitive Information Processing (CIP), 2012 3rd International Workshop on*, pages 1–6. IEEE, 2012.

[GVL96] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 1996.

[GW77] R. C. Gonzales and P. Wintz. *Digital Image Processing.* Addison-Wesley, Reading, MA, 1977.

[H+70] Walter C Hamilton et al. The revolution in crystallography. *Science (New York, NY)*, 169(3941):133, 1970.

[HAH11] Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. A randomized heuristic for kernel parameter selection with large-scale multi-class data. In *Proceedings of 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2011.

[HAH13] Toke Jansen Hansen, Trine Julie Abrahamsen, and Lars Kai Hansen. Improved kernel pca denoising by semi-supervised manifold learning. *Pattern Recognition Letters*, Under review, 2013.

[HAK+00] Alexander Hinneburg, Charu C Aggarwal, Daniel A Keim, et al. *What is the nearest neighbor in high dimensional spaces?* Bibliothek der Universität Konstanz, 2000.

[Her01] Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[HGF+01] James V Haxby, M Ida Gobbini, Maura L Furey, Alumit Ishai, Jennifer L Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.

[HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[HL02] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[HR04a] D Hoyle and Magnus Rattray. Limiting form of the sample covariance eigenspectrum in pca and kernel pca. *Advances in Neural Information Processing Systems*, 16:16–23, 2004.

[HR04b] David Hoyle and Magnus Rattray. A statistical mechanics analysis of gram matrix eigenvalue spectra. *Learning Theory*, pages 579–593, 2004.

[HR07] David C Hoyle and Magnus Rattray. Statistical mechanics of learning multiple orthogonal signals: asymptotic theory and fluctuation effects. *Physical Review E*, 75(1):016101, 2007.

[HR09] P. Honeine and C. Richard. Solving the pre-image problem in kernel machines: A direct method. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*, pages 1–6. IEEE, 2009.

[HR11] P. Honeine and C. Richard. Preimage problem in kernel-based machine learning. *Signal Processing Magazine, IEEE*, 28(2):77–88, 2011.

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36:1171–1220, 2008.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Science, 2001.

[Hub] Peter J Huber. Robust statistics. 1981.

[Hul94] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, 1994.

[Joa01] Thorsten Joachims. *The Maximum-Margin Approach to Learning Text Classifiers:Methods, theory, and algorithms*. PhD thesis, Dortmund University, 2001.

[JT09] Iain M Johnstone and D Michael Titterington. Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253, 2009.

[KFS05] Kwang In Kim, Matthias O. Franz, and Bernhard Schölkopf. Iterative kernel principal component analysis for image modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(9):1351–1366, 2005.

[KHR+13] Maya Kallas, Paul Honeine, Cedric Richard, Clovis Francis, and Hassan Amoud. Non-negativity constraints on the pre-image for pattern recognition with kernel machines. *Pattern Recognition*, 46(11):3066 – 3080, 2013.

[KHS01a] Ulrik Kjems, Lars K. Hansen, and Stephen C. Strother. Generalizable singular value decomposition for ill-posed datasets. In *Advances in Neural Information Processing Systems 13*, pages 549–555. MIT Press, 2001.

[KHS+01b] Ulrik Kjems, Lars K Hansen, Stephen C Strother, et al. Generalizable singular value decomposition for ill-posed datasets. *Advances in neural information processing systems*, pages 549–555, 2001.

[KM97] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proc. 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[KT04] James Tin-Yau Kwok and Ivor Wai-Hung Tsang. The pre-image problem in kernel methods. *IEEE transactions on neural networks*, 15(6):1517–1525, 2004.

[KW71] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[LdC08] Ana Carolina Lorena and André C. P. L. F. de Carvalho. Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomput.*, 71:3326–3334, October 2008.

[LP11] C. Leitner and F. Pernkopf. The pre-image problem and kernel pca for speech enhancement. *Advances in Nonlinear Speech Processing*, pages 199–206, 2011.

[LP12a] C. Leitner and F. Pernkopf. Musical noise suppression for speech enhancement using pre-image iterations. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pages 464–467. IEEE, 2012.

[LP12b] C. Leitner and F. Pernkopf. Speech enhancement using pre-image iterations. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4665–4668. IEEE, 2012.

[LP12c] C. Leitner and F. Pernkopf. Suppression of musical noise in enhanced speech using pre-image iterations, 2012.

[LP13] Christina Leitner and Franz Pernkopf. Generalization of pre-image iterations for speech enhancement. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, In Press, 2013.

[Mac98] David JC MacKay. Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

[Mer09] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Royal Soc. (A)*, 83(559):69–70, 1909.

[MFA+13] AF Marquand, M Filippone, J Ashburner, M Girolami, J Mourao-Miranda, GJ Barker, SCR Williams, PN Leigh, and CRV Blain. Automated, high accuracy classification of parkinsonian disorders: a pattern recognition approach - supplementary information. *PLoS ONE*, 2013.

[Mod89] Rudy Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3):233–248, 1989.

[MRW+99] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and KR Müllers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE, 1999.

[MSS+99] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems 11*, pages 536–542. MIT Press, 1999.

[Nie11] A.A. Nielsen. Kernel maximum autocorrelation factor and minimum noise fraction transformations. *Image Processing, IEEE Transactions on*, 20(3):612–624, 2011.

[NlTF08] M. H. Nguyen and F. De la Torre Frade. Robust kernel principal component analysis. In *Advances in Neural Information Processing Systems*, December 2008.

[Pla98] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.

[RAMH12] Peter Mondrup Rasmussen, Trine Julie Abrahamsen, Kristoffer Hougaard Madsen, and Lars Kai Hansen. Nonlinear denoising and analysis of neuroimages with kernel principal component analysis and pre-image estimation. *NeuroImage*, 60(3):1807 – 1818, 2012.

[RDT06] Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel pca. In *Proceedings of SPIE*, volume 6064, pages 425–432, 2006.

[RHM+12] Peter M Rasmussen, Lars K Hansen, Kristoffer H Madsen, Nathan W Churchill, and Stephen C Strother. Model sparsity and brain pattern interpretation of classification models in neuroimaging. *Pattern Recognition*, 45(6):2085–2100, 2012.

[RK04] Bhavani Raskutti and Adam Kowalczyk. Extreme re-balancing for svms: a case study. *ACM Sigkdd Explorations Newsletter*, 6(1):60–69, 2004.

[RS00] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[RW64] J.R. Rice and J.S. White. Norms for smoothing and estimation. *SIAM review*, 6(3):243–256, 1964.

[SAH+02] Stephen C Strother, Jon Anderson, Lars Kai Hansen, Ulrik Kjems, Rafal Kustra, John Sidtis, Sally Frutiger, Suraj Muley, Stephen LaConte, David Rottenberg, et al. The quantitative evaluation of functional neuroimaging experiments: The npairs data analysis framework. *NeuroImage*, 15(4):747–771, 2002.

[SCB13] A.K. Seth, P. Chorley, and L.C. Barnett. Granger causality analysis of fmri bold signals is invariant to hemodynamic convolution but not downsampling. *NeuroImage*, 65:540–555, January 2013.

[SMB+99] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions On Neural Networks*, 10(5):1000–1017, 1999.

[SNB05] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 824–831. ACM, 2005.

[SR03] L.K. Saul and S.T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

[SS01] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[SS04] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[SSWB00] Bernhard Schlökopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Comput.*, 12(5):1207–1245, 2000.

[STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[STC+08] João Ricardo Sato, Carlos Eduardo Thomaz, Ellison Fernando Cardoso, André Fujita, Maria da Gra a Morais Martin, and Edson Amaro. Hyperplane navigation: a method to set individual scores in fmri group datasets. *Neuroimage*, 42(4):1473–1480, 2008.

[STW03] John Shawe-Taylor and Christopher KI Williams. The stability of kernel principal components analysis and its relation to the process eigenspectrum. *Advances in neural information processing systems*, 15:367–374, 2003.

[SZD13] M.E. Salih, X.M. Zhang, and M.Y. Ding. An improvement of non-local means denoising method in the presence of large noise. *Applied Mechanics and Materials*, 263:223–226, 2013.

[T+98] J.B. Tenenbaum et al. Mapping a manifold of perceptual observations. *Advances in neural information processing systems*, pages 682–688, 1998.

[TDSL00] J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[TFX12] Zhiying Tan, Yong Feng, and Kejia Xu. A local linear algorithm for calculating the pre-image in kernel pca. In *Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on*, pages 100–103, 2012.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[TK02] Takashi Takahashi and Takio Kurita. Robust de-noising by kernel pca. In *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, pages 739–744. Springer-Verlag, 2002.

[TMA+12] M. Tangermann, K.R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K.J. Miller, G.R. Müller-Putz, et al. Review of the bci competition iv. *Frontiers in Neuroscience*, 6, 2012.

[TTSL08] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, and E. W. Lang. Kpca denoising and the pre-image problem revisited. *Digit. Signal Process.*, 18(4):568–580, 2008.

[TZCK09] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):281–288, 2009.

[Vap79] Vladimir Naumovich Vapnik. Estimation of dependencies based on emperical data [in russian]. *Nauka, Moscow*, 1979. (English translation: Springer-Verlag, New York, 1982).

[Vap95] Vladimir Naumovich Vapnik. *The Nature of Statistical Learning*. Springer-Verlag, New York, 1995.

[Vap98] Vladimir Naumovich Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc, New York, 1998.

[VC71] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[VC00] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Comput.*, 12:2013–2036, September 2000.

[VHB10] CJ Van Heerden and E. Barnard. Towards understanding the influence of svm hyperparameters. In *21st Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, pages 69–74, 2010.

[VM11] Matthias Varewyck and Jean-Pierre Martens. A Practical Approach to Model Selection for Support Vector Machines With a Gaussian Kernel. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(2):330–340, April 2011.

[Wah99] Grace Wahba. *Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV*, pages 69–88. MIT Press, Cambridge, MA, USA, 1999.

[WC03] Gang Wu and Edward Y Chang. Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, pages 49–56, 2003.

[Wei04] Gary M Weiss. Mining with rarity: a unifying framework. *Sigkdd Explorations*, 6(1):7–19, 2004.

[WHMH10] Christian Walder, Ricardo Henao, Morten Mørup, and Lars Kai Hansen. Semi-supervised kernel pca. *CoRR*, abs/1008.1398, 2010.

[Wil98] Christopher KI Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:599–621, 1998.

[WMC+01] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. *Advances in neural information processing systems*, pages 668–674, 2001.

[WR96] CKI Williams and CE Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 1996.

[WS04] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 988–995. IEEE, 2004.

[WSS04] K.Q. Weinberger, F. Sha, and L.K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, pages 839–846. ACM, 2004.

[WW09] Kuo-Ping Wu and Sheng-De Wang. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recognition*, 42:710–717, May 2009.

[XXCJ10] Song Xiaoshan, Jiang Xiaoyu, Han Chongzhao, and Luo Jianhua. Inter-class distance based kernel parameter evaluating method for rbf-svm. *Digital Manufacturing and Automation, International Conference on*, 1:853–858, 2010.

[ZB06] Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. *Advances in neural information processing systems*, 18:1649–1656, 2006.

[ZL06] W.S. Zheng and J. Lai. Regularized locality preserving learning of pre-image problem in kernel principal component analysis. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 456–459. IEEE, 2006.

[ZLY06] W.S. Zheng, J.H. Lai, and P.C. Yuen. Weakly supervised learning on pre-image problem in kernel methods. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 711–715. IEEE, 2006.

[ZLY10] W.S. Zheng, J.H. Lai, and P.C. Yuen. Penalized preimage learning in kernel principal component analysis. *Neural Networks, IEEE Transactions on*, 21(4):551–570, 2010.